Universidade de Aveiro
2014

Departamento de Eletrónica,
Telecomunicações e Informática

ANDRÉ
BRANQUINHO
GOMES

OPTIMIZATION OF THE ELECTRICAL POWER DISTRIBUTION IN A MODERN AIRCRAFT

OPTIMIZAÇÃO DA DISTRIBUIÇÃO DE POTÊNCIA ELÉCTRICA NUM AVIÃO MODERNO

Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2014

ANDRÉ BRANQUINHO GOMES

OPTIMIZATION OF THE ELECTRICAL POWER DISTRIBUTION IN A MODERN AIRCRAFT

OPTIMIZAÇÃO DA DISTRIBUIÇÃO DE POTÊNCIA ELÉCTRICA NUM AVIÃO MODERNO

This master thesis was performed as part of a project in SILVER ATENA Electronic Systems Engineering GmbH, under the supervision of Prof. Dr. Anusch Taraz, head of the chair Discrete Mathematics, Institute of Mathematics at the Hamburg University of Technology. As a student of the University of Aveiro, this work was also co-supervised by Prof. Dr. Amaro Fernandes de Sousa, assistant professor of the Department of Electronics, Telecommunications and Informatics at the University of Aveiro.

To my dear Maria and to all our family

the jury

First Examiner — Prof. Dr. Anusch Taraz
Hamburg University of Technology

Second Examiner — Prof. Dr. Amaro Fernandes de Sousa
Universidade de Aveiro

acknowledgments

I would like to thank SILVER ATENA for the opportunity of doing this master thesis in such an interesting field as the aircraft industry. I want to express my deepest gratitude to Prof. Dr. Anusch Taraz, without whom I would have never been able to take advantage of this opportunity. His skillful guidance, tranquility and understanding were essential in the completion of this work. I would also like to express my sincere appreciation to Prof. Dr. Amaro de Sousa, who always made himself available to support this work. All the conversations and advice were extremely helpful for the good course of this thesis.

I am very grateful to Dipl.-Ing. Oliver Götting and B.Sc. Bernd Busam for guiding my work in SILVER ATENA and for the help with reviewing this thesis. They were always available to help me overcome the various difficulties I have encountered over the past few months.

I would like to thank Dipl.-Ing. Heiko Fimpel, Dr. Christian Müller and Dipl.-Math. Wolfgang Riedl for their useful suggestions.

I would like to extend my gratitude to B.Eng. Musa Cevahir, Dipl.-Ing. (FH) Lars Liebegut and Dipl.-Ing. Wolfgang Salomon for all the companionship in the office.

I would like to thank Rui for keeping me company when I had to work, Sandro for all the help he gave me here in Hamburg and Soraia for all the really nice meals she prepared. Without them life here in Hamburg would have been far less happy.

I would like to deeply thank my mother and father, who always did everything they could and could not do to support me. I want also to profoundly thank my dear (and soon to be wife) Maria. Her encouragement over the past few years was fundamental for the completion of my degree, and particularly for coming to Hamburg. All the support and love of our family were essential every single day of this thesis.

keywords aircraft, electrical power, optimization, mixed-integer linear programming.

abstract

The technical progress of devices for the aircraft sector, as well as the desire for a more comfortable and pleasant flight, are constantly increasing. Due to this, more and more electrical equipments, not essential for the basic operation of the aircraft, are installed. These can be in-flight entertainment systems, power supplies to portable electronic devices, among several others. Most of these electrical devices are spread over cabin and cargo and must be connected to the power system of the aircraft.

In order to satisfy the different demands of airline companies, aircraft manufacturers allow an increasing customization capability in their aircrafts, and different combinations of devices can be chosen for installation. Redesigning and optimizing the distribution for each specific aircraft order would be expensive and time-consuming. As a result, a fixed system is used and the connections of the customizable devices must then comply with several criteria.

The different types of equipments, along with the variation of their power consumption during flight, lead to difficult manual connection decisions. Thus, the main target becomes finding a viable connection scheme and not necessarily an optimal one. This suggests possible margins for improvements.

This thesis focuses on the automation of these decisions, by optimizing them according to different objectives, such as three-phase power balance and weight. The modeling of the problem is based on Mixed-Integer Linear Programming and an optimization tool is developed using MATLAB®.

The results for the test cases show direct improvements in the system's parameters with considerably lower time effort. Most of the optimizations are performed within a few minutes and may lead to substantial improvements in the total customization time. Moreover, the results also demonstrate the potential of these optimizations to enhance the future cabin and cargo power system architectures.

The final software automates the allocation decisions and includes possible specifications, such as optimization targets (single or multiple), and partial or full system optimization, as well as other parameters.

palavras chave

avião, potência elétrica, optimização, programação linear inteira mista.

resumo

O desenvolvimento tecnológico dos dispositivos utilizados na indústria aeronáutica, bem como o desejo de proporcionar um vôo mais confortável e agradável, estão em constante progressão. Consequentemente, cada vez mais equipamentos eléctricos, não essenciais para o funcionamento básico da aeronave, estão a ser instalados. Estes podem ser sistemas de entretenimento a bordo, fontes de alimentação para dispositivos portáteis, ou outros. Estes dispositivos são na sua maioria distribuídos pela cabine e pelo porão e têm de ser ligados ao sistema eléctrico da aeronave.

De modo a satisfazer as diferentes exigências das companhias aéreas, os fabricantes de aviões possibilitam uma crescente personalização das suas aeronaves, pelo que diferentes combinações de dispositivos podem ser escolhidas para instalação. Redesenhar e optimizar o sistema de distribuição eléctrica para cada avião encomendado seria caro e demorado. Por esse motivo, é utilizado um sistema de distribuição fixo e as ligações dos dispositivos personalizáveis têm de cumprir vários critérios.

Os diferentes tipos de equipamentos, tal como a sua variação de consumo energético durante o vôo, tornam difíceis as decisões de ligação feitas manualmente. Por esta razão, nas decisões manuais, o principal objetivo é encontrar um esquema de ligações viável e não necessariamente óptimo. Isto indicia a existência de margem para melhorias.

O principal objetivo desta tese é a automatização destas decisões, através da sua optimização de acordo com diferentes objectivos, tais como o balanceamento de potência trifásica e o peso. A modelação do problema é baseada em Programação Linear Inteira Mista e é desenvolvida uma ferramenta de optimização utilizando o MATLAB®.

Os resultados para os casos de teste demonstram uma melhoria directa nos parâmetros do sistema, com considerável diminuição do tempo despendido. A maior parte das optimizações são realizadas em alguns minutos, o que pode levar a melhorias substanciais no tempo total de personalização da aeronave. Além disso, os resultados revelam também o potencial destas optimizações para melhorar as futuras arquiteturas do sistema de distribuição eléctrica de cabine e porão.

O software final desenvolvido automatiza as decisões de ligação e permite incluir várias especificações, tais como objectivos de optimização (únicos ou múltiplos), optimização de parte ou da totalidade do sistema, entre outros parâmetros.

Contents

List of Figures

List of Tables

Glossary

A/C	Aircraft
AC	Alternate Current
APU	Auxiliary Power Unit
DC	Direct Current
ELMS	Electrical Load Management System
GPU	Ground Power Unit
IFE	In-Flight Entertainment
LP	Linear Programming
IP	Integer Linear Programming
BIP	Binary Integer Linear Programming
MCR	MATLAB Compiler Runtime
MIP	Mixed-Integer Linear Programming
LRM	Line Replaceable Module
OR	Operations Research
P/CD	Protection and Commutation Device
PEPDC	Primary Electrical Power Distribution Center
PM	Power Management
PMG	Permanent Magnet Generator
RAT	Ram Air Turbine
RCCB	Remote Control Circuit Breaker
SEPDC	Secondary Electrical Power Distribution Center
SPDB	Secondary Power Distribution Box
SSPC	Solid State Power Controller
TRU	Transformer Rectifier Unit
VBA	Visual Basic for Applications
VFG	Variable Frequency Generator

CHAPTER 1

Introduction

The technical progress of devices for the aircraft sector, as well as the desire for a more comfortable and pleasant flight, are constantly increasing. Due to this, more and more electrical equipments not relevant for the basic operation of the aircraft, also known as commercial loads, are installed. These can be in-flight entertainment systems, power supplies to portable electronic devices, sophisticated lighting and floor panel heaters, or even electrical cargo loading systems that promote better working conditions. Hence, these electrical consumers are not only used to enhance the passenger's experience, but also to assist cabin and airport crew operations [31].

These electrical devices are spread over cabin and cargo and must be connected to the power distribution system of the aircraft. Some of them are standard and some are optional. The standard devices are part of every aircraft and have already predefined connection schemes. However, in order to satisfy the different demands of the airline companies, aircraft manufacturers allow an increasing customization capability in their aircrafts. As a result, additional optional devices can be connected to the power distribution system.

This poses an extra challenge to the aircraft manufacturer. The power distribution system cannot be fully redesigned and optimized for each specific aircraft order, since this would be expensive and time-consuming. Therefore, a standardized network is used, where only minor changes are allowed. Although this is more economical in terms of production, it leads to constraints such as power limitations.

In order to be able to respond to most of the customization demands, as well as further changing possibilities, margins are added to the network. Still, they do not fully eliminate the constraints imposed by the fixed architecture, therefore the connections of the optional loads must comply with multiple criteria.

1.1 Motivation

A representative structure of the electrical distribution network of various modern aircrafts is depicted in Fig. 1.1. After generation, the power is first delivered to the primary distribution center below the cockpit. Electrical loads with high power demands are connected directly to the primary distribution center, while the remaining are allocated either to secondary distribution centers or secondary distribution boxes. Almost all of the commercial loads are connected to the secondary distribution boxes, located in the cabin and cargo. These boxes and all the power lines upstream (towards the source) and downstream (towards the loads) can be referred to as the cabin and cargo distribution [31].

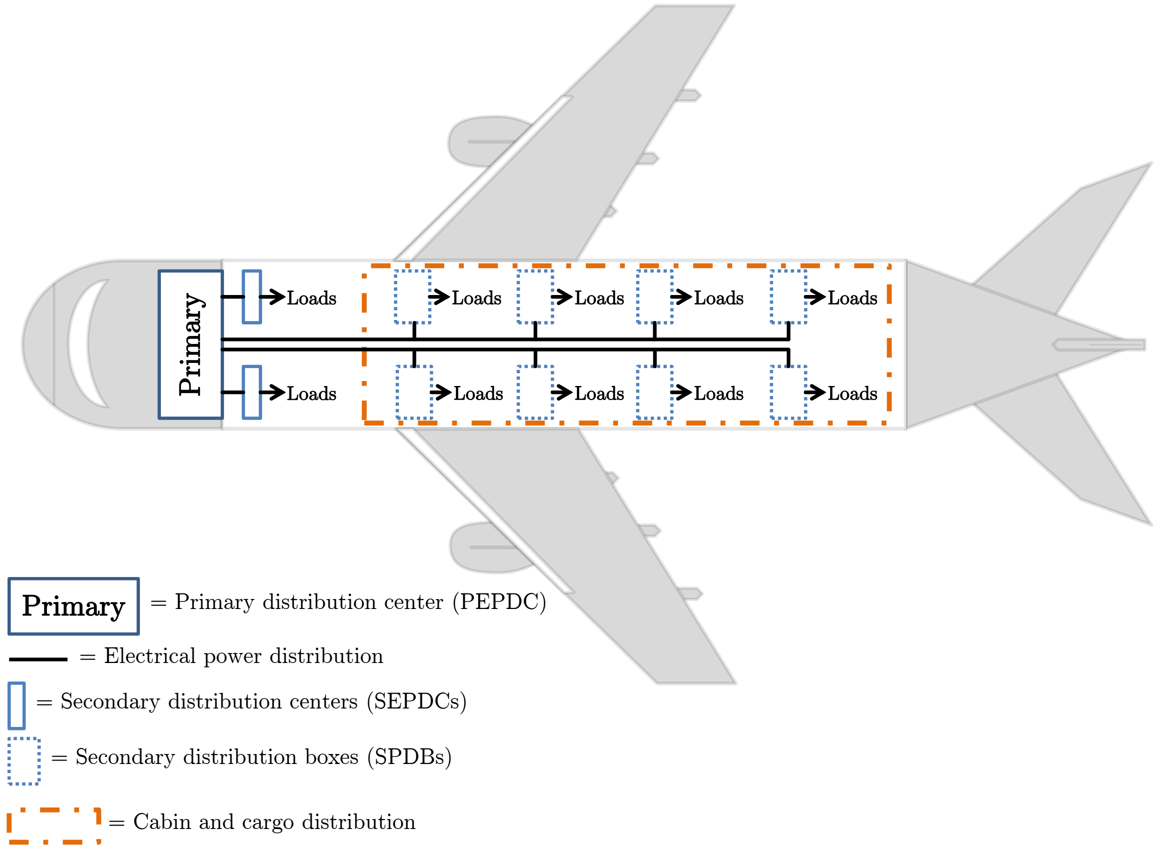


Figure 1.1: Structure of the electrical network of various modern aircrafts (adapted from [31]).

Fig. 1.2 shows a scheme of the complete power distribution system, with emphasis on the cabin and cargo distribution. Each Secondary Power Distribution Box (SPDB) is supplied by a variable number of feeders (alternate current (AC) and direct current (DC)). These feeders supply different Line Replaceable Modules (LRMs), which consist of groups of Solid State Power Controllers (SSPCs). Finally, each load is allocated to one SSPC. These loads can represent a single device or a group of devices that can be supplied together. In either case, they are regarded as a single load [31], [36].

The SPDBs are spread inside cabin and cargo in order to supply loads in different locations. To minimize the length and thickness of the downstream cables, each load is usually connected to the nearest SPDB. Nevertheless, inside each SPDB, the loads can be allocated to different SSPCs, which are supplied by distinct feeders. This depends on the LRM where the SSPC is located. Furthermore, in the case of AC loads, the feeders contain three different cables, supplying each of the electrical phases (A,B and C). Hence, the SSPC allocation choices have an impact not only on the power on each feeder, but also on the power on each electrical phase.

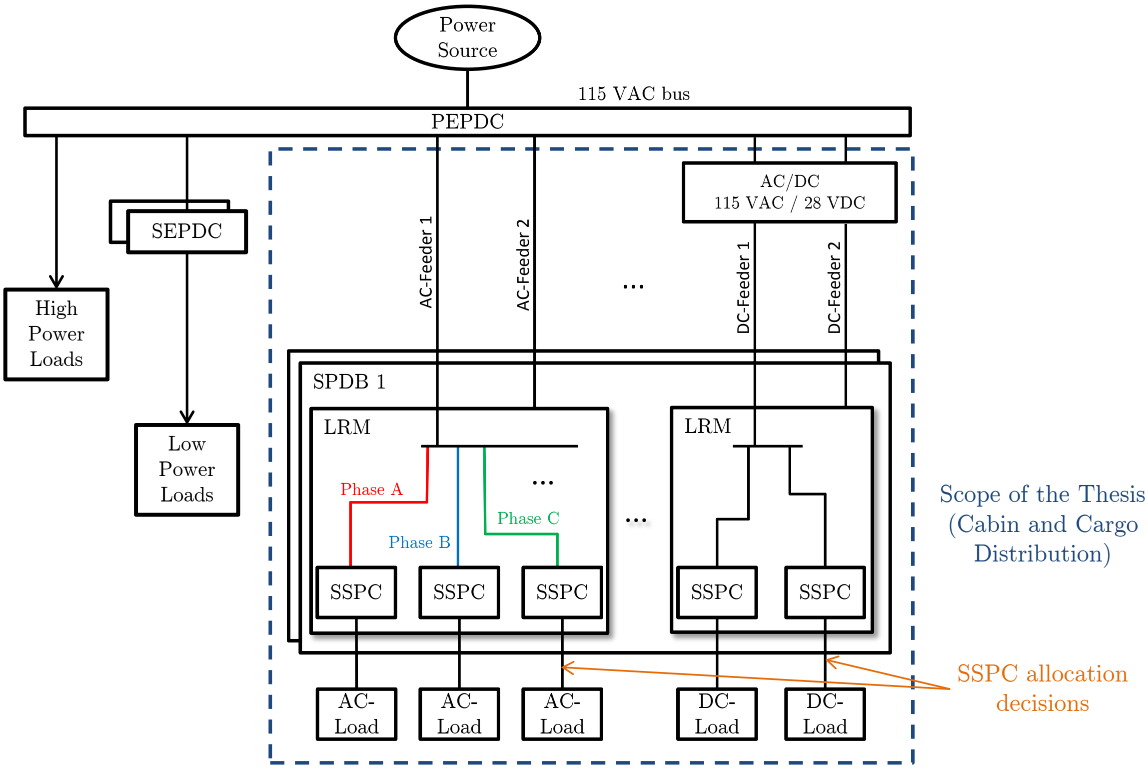


Figure 1.2: Power distribution system scheme with emphasis on the cabin and cargo distribution (adapted from [36] and [31]).

The different types of customizable electrical loads, along with the variation of their power consumption during flight, lead to difficult manual allocation decisions. As a result of the procedure's high complexity, the main target of customization becomes to find a viable allocation scheme, and not an optimal one. This suggests margins for improvements.

This thesis focuses on the study of the optional loads' connection choices. Looking again at Figure 1.2, this corresponds to deciding the SSPC where each optional load is allocated. These decisions may affect the power system's performance and possibly lead to enhancements in the architecture of the distribution network. Due to its complexity, automating the decisions can also be important to reduce the effort and time involved in this process.

1.2 OBJECTIVES

The main objective of this work is to develop a program that automates the allocation decisions, according to defined optimization targets.

For this purpose, the cabin and cargo distribution system must be analyzed in detail. This step is necessary to fully understand the problem in study and also to propose possible optimization objectives.

In order to develop the program for automating the decisions, and evaluate the impacts of the different optimizations, a suitable model of the system must be developed.

The validity of the obtained solutions is tested with the help of previously developed software, and their quality is assessed based in comparison with manual allocations. Two main criteria shall be considered: (1) the quality of the solution with respect to one or more parameters, and (2) the time needed for completing the allocations.

Finally, a complete tool should be developed. This includes all the necessary steps that enable the prompt usage of the optimization procedure integrated with the remaining software.

1.3 STRUCTURE

This thesis is organized as follows: Chapter 2 introduces the electrical system of an aircraft, with emphasis on the secondary power distribution to the commercial loads. It intends to give the reader the basis for understanding the system to be optimized and explain in more detail the various characteristics that have to be included in the development of the model.

In Chapter 3, an overview of optimization is given. After a brief historical introduction, the most important concepts are defined, and the necessary optimization methods for this thesis are analyzed.

With the system and optimization concepts properly defined, Chapter 4 provides a discussion of the possible objectives of optimization and a more systematized overview of the conditions that the system must verify.

In Chapter 5, we look in detail to the mathematical models created, regarding all the possibilities previously described. The implementation in MATLAB® is explained in Chapter 6. Both these chapters aim to outline the necessary concepts for performing the optimization of the system.

Some of the results of this thesis are presented and discussed in Chapter 7. These results include the most common situations of optimization. This is the chapter where the efficiency of the developed model is analyzed, and the improvements to the prior customizations are measured.

The final tool for the customization procedure is presented in Chapter 8. It takes into account the demanded features and some of the results obtained.

Finally, the conclusions of this work are outlined in Chapter 9, as well as possible future work.

CHAPTER 2

System Description

In this section, a brief overview of the electrical system of a modern aircraft (A/C) is given, and the main points of power generation and distribution are described. For this purpose, an aircraft model similar to the one presented in [31] will be used. However, it shall be noted that this example is not intended to precisely define a specific aircraft, but only to give the reader a general idea.

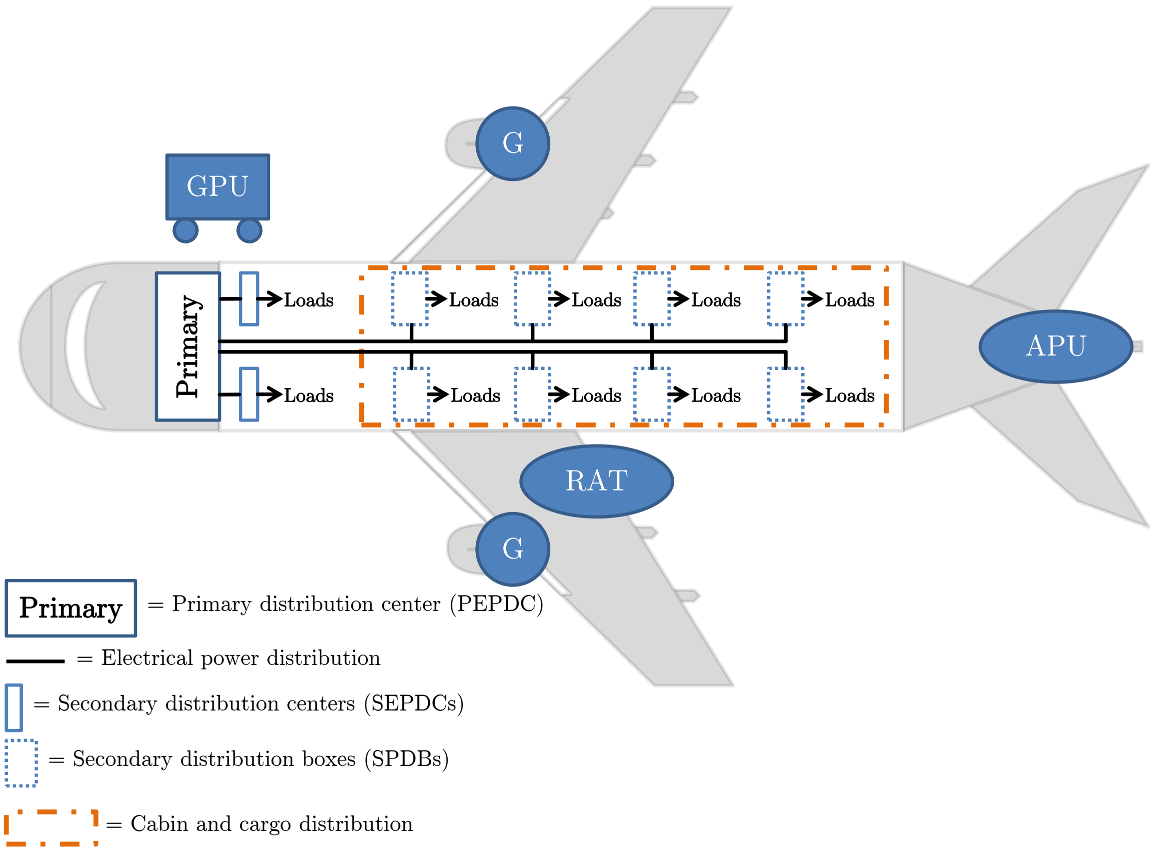


Figure 2.1: Structure of the electrical network of various modern aircrafts (adapted from [31]).

2.1 ELECTRICAL SYSTEM

The aircraft's electrical system comprises the following three functions [31]:

- electrical power generation,
- electrical power distribution, and
- electrical loads' supply.

Protection, monitoring and load management are some of the remaining functions the electrical system has to perform. As an example of a modern aircraft, we can look at Figure 2.1. The aircraft has two generators, each located in one of the two main engines. As it will be described later, they generate AC three-phase power, with a specified line-to-ground nominal voltage. All of the generators supply their respective network, for segregation reasons. Another two generators are attached to the Auxiliary Power Unit (APU) , located in the back. A Ram Air Turbine (RAT) supplies the AC network if all the other generators are lost during flight. Ground Power Units (GPUs) provide electrical power to the A/C on ground [31].

Regardless of the source, the electrical power is first delivered to the Primary Electrical Power Distribution Center (PEPDC) , located below the cockpit. Systems with high power demands, that is, which require currents greater than 15 A, are directly connected to the PEPDC. Loads with required currents below 15 A are either supplied by Secondary Electrical Power Distribution Centers (SEPDCs) or by Secondary Power Distribution Boxes (SPDBs). These secondary distribution systems are supplied by the PEPDC and have different loads attached. The SEPDCs supply light loads essential for the basic operation of the aircraft, also called essential loads. The SPDBs supply the previously mentioned commercial loads, also called non-essential loads [31]. The latter are located in the cabin and cargo and are not necessary for the basic operation of the aircraft. They are mainly used to enhance the passengers' experience and assist personnel.

Although it is not relevant for the purpose of this thesis, it should be said that an emergency power center is used when all the other power centers are lost, and supplies only emergency loads to allow the safe completion of the flight.

The AC power generation and distribution play an important role in the operation of the aircraft, supplying lighting, In-Flight Entertainment (IFE) systems and AC motors (among several others).

In the aircraft, three-phase AC power is used (DC power is generated from the AC, as will be described later). Some of the general advantages of this type of power distribution, when compared to single-phase, are: (1) the possibility of having constant instantaneous power (not pulsating), which results in uniform power transmission and consequent less vibration and abrasion of machines, and (2) the more economical transmission of a given amount of power [1].

Despite the advantages of power transmission when compared to single-phase AC power distribution, special attention needs to be paid to the balancing of the three phases. Unbalance between the electrical phases can have negative impact on the generator, on the wiring and even on the loads [19].

2.1.1 GENERATION

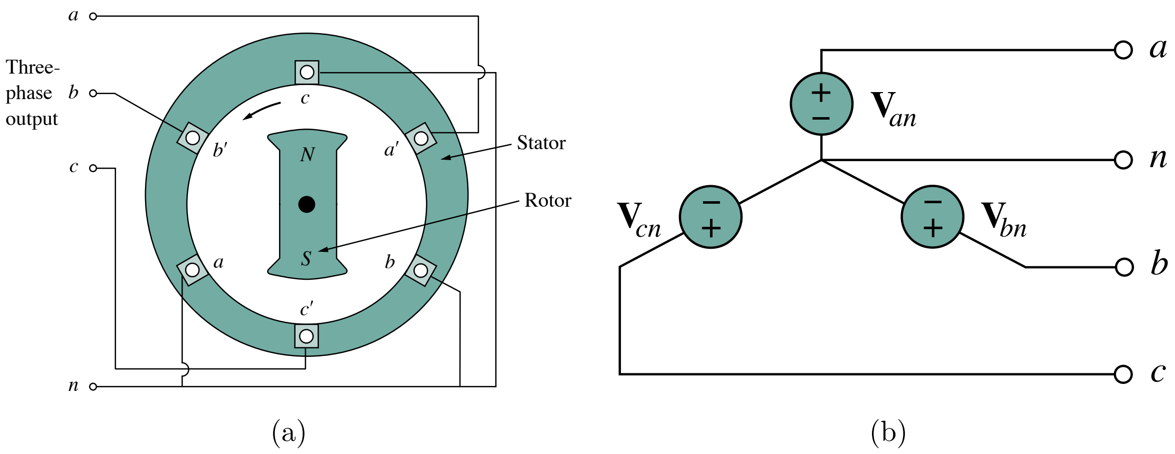


Figure 2.2: Three-phase AC power generation: (a) alternator (taken from [1]); (b) output (taken from [1]).

Three-phase voltages are usually produced with a three-phase AC generator, also referred to as alternator. Figure 2.2a depicts a cross-sectional view of a basic alternator. This consists of a rotating magnet (called *rotor*) surrounded by a stationary winding (called *stator*). Three separate wirings are placed around the stator, 120° apart. The voltage of each turn of the winding is out of phase with the voltage generated in its neighbor, since it is cut by maximum magnetic flux density an instant earlier or later. Due to the position of the wirings, the voltages' phase difference is equal to 120°, but their magnitude is the same for every coil. Since each coil can be regarded as a single-phase generator on its own, this generator can supply both single-phase and three-phase loads [1], [11].

The output of the generator can be represented as in Figure 2.2b. The voltages V_{an}, V_{bn} and V_{cn} (voltages between lines a, b, and c, and the neutral line n) can be described as

$$V_{an} = V_m \cos \omega t \tag{2.1}$$
$$V_{bn} = V_m \cos(\omega t - 120°) \tag{2.2}$$
$$V_{cn} = V_m \cos(\omega t - 240°), \tag{2.3}$$

where V_m is the maximum voltage and ω is the radian frequency. Using phasor notation (for more information on phasor notation, please refer to Appendix B):

$$\boldsymbol{V}_{an} = V_m \angle 0° \tag{2.4}$$
$$\boldsymbol{V}_{bn} = V_m \angle {-120°} \tag{2.5}$$
$$\boldsymbol{V}_{cn} = V_m \angle {-240°}. \tag{2.6}$$

The three voltages are said to be balanced because they have the same amplitude V_m, the same frequency ω, and are out of phase with each other by exactly 120°.

Modern aircrafts such as the one in analysis, use a Variable Frequency Generator (VFG) , which is a three-stage AC generator. The first stage is a Permanent Magnet Generator (PMG) . The second is a main exciter with a full-wave rectifier that receives its exciter field from the PMG. The third provides power to the aircraft electrical network [31]. The variable frequency included in the name of the generator comes from the fact that the resulting network frequency can vary typically from 380 to 720 Hz. To solve possible issues arising from this variation, major aircraft manufacturers place the burden upon equipment suppliers to ensure that their systems work over the specified range of frequencies [24].

The main characteristic of AC power generation is that it operates at a higher voltage, normally 115 VAC, instead of 28 VDC for the DC system. Higher voltage requires better insulation standards.

Yet, for a given power value, higher voltage enables a lower current. The lower the current, the lower the power losses and the lighter the cables [24]. This is one of the reasons supporting 115 VAC generation and distribution, instead of 28 VDC.

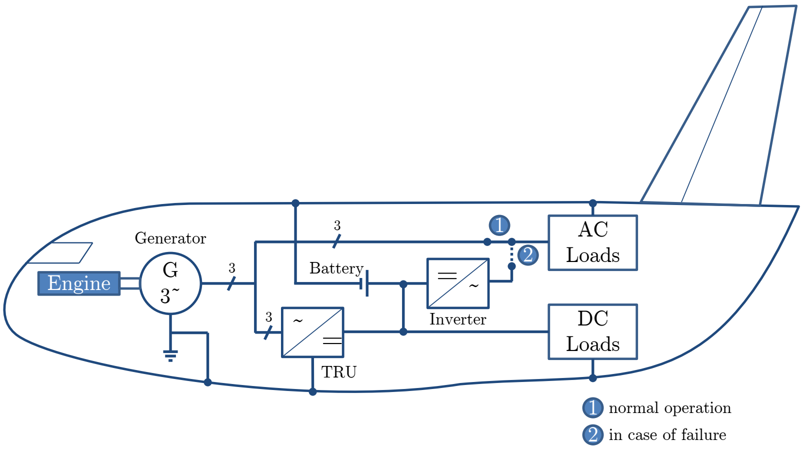


Figure 2.3: Simplified model of the electrical power generator circuit (adapted from [31]).

Still, the aircraft also contains DC loads, so DC power must be generated. The normal conversion to DC power is achieved using Transformer Rectifier Units (TRUs), as depicted in Figure 2.3. Here, a simplified model of a generator circuit is shown. The generator is attached to the engine, producing AC power. The AC power is transformed into DC with the help of a TRU, which provides the desired 115 VAC/28 VDC conversion [24]. If no AC power is available (in case of failure), batteries supply DC loads and the AC loads are supplied using an inverter [31].

2.1.2 TRANSMISSION

As previously stated, three-phase AC systems are usually more economical than single AC systems. While a single AC system needs two conductors, a three-phase system delivers three times more power and only needs two more wires: three phases and neutral (depending on the type of connection, sometimes only a total of three wires are needed). That is, a 200% increase in power transmission is done with a cost increase of about 100%. The improvement in weight (due to the lower total number of cables) is an important aspect to consider, when projecting an aircraft.

The output of the three-phase AC generator is connected to the loads by means of transformers and transmission lines. Nevertheless, to analyze such a circuit, we can reduce it to a voltage source connected to a load via a line (see Figure 2.4), without jeopardizing the desired conclusions of the analysis [27].

There are two basic three-phase configurations for the generator: delta (Δ) and wye (Y). The simple scheme depicted in Figure 2.2b is in a Y configuration. This was not a coincidence: in an aircraft the three phases are most often connected in Y configuration with one end of each of the phases connected to a neutral point [24]. Hence, this will be the chosen configuration for the circuit analysis. Like the generator, a three-phase load can also be either Y-connected or Δ-connected. Therefore, we have a total of four possible connections (considering also the two possibilities for the generator): Y-Y, Y-Δ, Δ-Δ and Δ-Y. The purposes of this analysis are independent of the configuration. For the reason mentioned before, this analysis will be done with a Y-Y connection. Furthermore, any balanced three-phase system can be reduced to an equivalent Y-Y system [1].

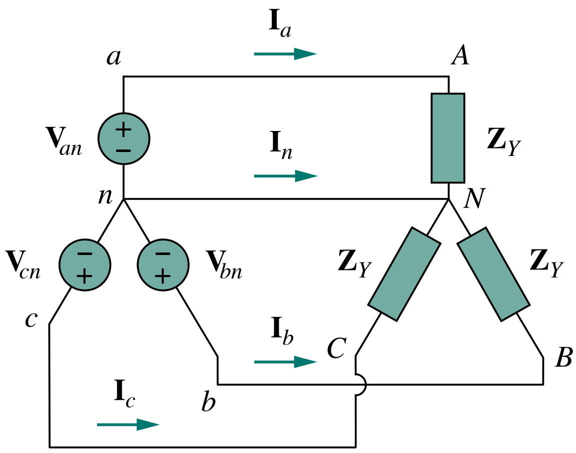


Figure 2.4: Three-phase AC power distribution analysis in a Y-Y configuration (taken from [1]).

Let us consider the basic circuit of Figure 2.4, where three equivalent loads Z_Y are connected to the AC generator (these can represent, for example, a three-phase load). Assuming the phase voltages defined in Equations (2.4)-(2.6) we can apply Kirchoff's voltage law to obtain the currents in each line:

$$\boldsymbol{I}_a = \frac{\boldsymbol{V}_{an}}{\boldsymbol{Z}_Y} \tag{2.7}$$

$$\boldsymbol{I}_b = \frac{\boldsymbol{V}_{bn}}{\boldsymbol{Z}_Y} = \frac{\boldsymbol{V}_{an}\angle{-120^\circ}}{\boldsymbol{Z}_Y} = \boldsymbol{I}_a\angle{-120^\circ} \tag{2.8}$$

$$\boldsymbol{I}_c = \frac{\boldsymbol{V}_{cn}}{\boldsymbol{Z}_Y} = \frac{\boldsymbol{V}_{an}\angle{-240^\circ}}{\boldsymbol{Z}_Y} = \boldsymbol{I}_a\angle{-240^\circ}. \tag{2.9}$$

It can readily be inferred that

$$\mathbf{I}_a + \mathbf{I}_b + \mathbf{I}_c + \mathbf{I}_n = 0, \tag{2.10}$$

and consequently

$$\mathbf{I}_n = -(\mathbf{I}_a + \mathbf{I}_b + \mathbf{I}_c) = 0. \tag{2.11}$$

This means that no current flows in the cable connecting the neutrals of the generator and the load. In addition, current flowing in one phase is equal to the inverse of the sum of the currents flowing in the other phases. Thus, each conductor acts as the return path for the currents from the other two conductors. This happens only if the loads connected to each electrical phase are exactly the same. In this case we say that the system is balanced. As we will discuss later, problems arise when loads differ.

As previously described, the electrical power is transmitted from the generators to the PEPDC. The electrical networks are segregated, with each generator feeding one main bus bar. According to their location inside the aircraft (and also due to safety measures), loads can be either connected to side 1 (A/C left side) or side 2 (A/C right side) [31]. As it has already been explained, high-current loads and heavy loads essential to the aircraft operation are connected directly to the PEPDC, while low-current essential loads are connected to SEPDCs. Finally, non-essential loads, the focus of this work, are connected to SPDBs. Figure 2.5 shows a scheme illustrating these connections.

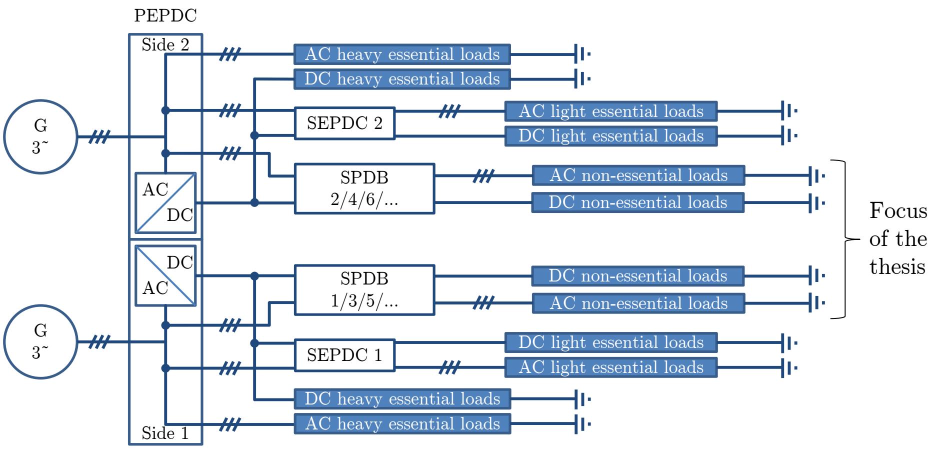


Figure 2.5: Transmission of the electrical power to the different types of loads (adapted from [31]).

2.1.3 SECONDARY DISTRIBUTION TO COMMERCIAL LOADS

DISTRIBUTION

Power lines, known as feeders, connect the PEPDC to the different SPDBs. There are several AC and DC feeders supplying the SPDBs. The SPDBs connected to side 1 are defined by an odd number (1,3,5,...), while SPDBs connected to side 2 have even numbers (2,4,6,...) [31]. In this thesis, the feeders will be identified by a number (even or odd, based on the sides described above) followed by "AC" or "DC", depending on the type of power. This is depicted in Figure 2.6.

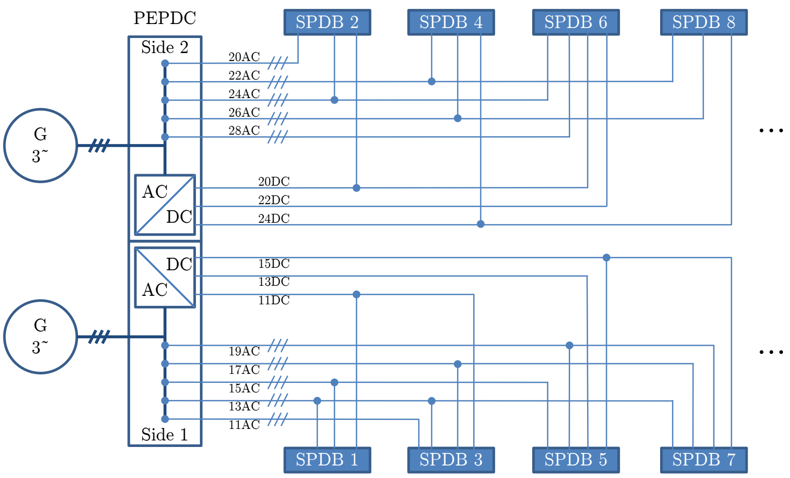


Figure 2.6: Example of a distribution network: feeders and cable segments (adapted from [31]).

This is the network responsible for supplying the non-essential loads, therefore the feeders must possess power margins that enable load customization by the airline companies. As it will be shown later, there are some scenarios where a fixed network is considered, and others where minor changes are allowed.

The on-essential loads can be separated in standard and optional. The tandard loads are part of every aircraft and have predefined allocations. The optional loads come from the possible customization of cabin and cargo.

The optimization of the optional loads' allocation choices is the motivation of this thesis, so the constraints coming from these networks will limit the possible choices (these constraints also involve the standard loads' connections). The allocation decisions will have impact on the network and this is the impact we want to study to be able to make the best decisions.

PROTECTION CONCEPT

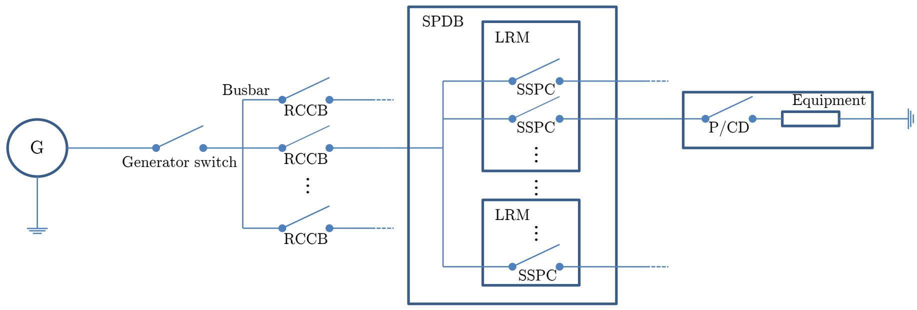


Figure 2.7: Protection devices of the secondary non-essential power distribution (adapted from [31]).

To avoid damage to the generators, the transmission network and the loads, protective devices are used in the aircraft electrical system. A scheme showing the devices used in the non-essential loads' power supply is depicted in Figure 2.7. Starting from the left, a main switch can disconnect the generator from its circuit, in case of overloads that cannot be cleared in lower levels or by load management functions. The feeders supplying the SPDBs are protected by Remote Control Circuit Breakers (RCCBs). As shown in Figure 1.2, SSPCs are located inside the SPDBs. The SSPCs are responsible for connecting the loads to the feeders and for protecting the wires between the SPDB and the loads. If an electrical fault occurs within the electrical loads, Protection and Commutation Devices (P/CDs), installed inside, open the circuit [31].

We will focus our attention on the RCCBs and SSPCs, since these are the ones that influence the connections of the loads as well as the dimensioning of the secondary power distribution network. RCCBs and SSPCs have replaced traditional circuit breakers and both are available with different ratings. In addition, each SSPC has a set of values and its choice is software programmable [7], [18], [33], [36]. Both use an electronically realized tripping curve. If electronics fail to open the circuit, due to a malfunction, a fail-safe unit is designed to open the line. The tripping and the fail-safe curve are placed between the maximum current of the load and the lowest no-damage curve of the wires connected. They also offer remote switching, which is very useful when load and power management are used [31].

POWER MANAGEMENT

Power Management (PM) techniques have been operated for about a decade on board some modern aircrafts [31]. They target the supply of personal electronic devices and this process enables more power outlets to be installed than the distribution network can actually supply. The basic idea is to suspend unused sockets, when a threat of overload appears. If this does not solve the problem, some sockets that are in use are turned off, and a battery takes over the supply.

Other power management techniques focus not only in switching off loads, but also on trying to reconfigure them in such a way that there is an optimal usage of power at all times. This is the case of the Electrical Load Management System (ELMS) [24].

More recently, motivated by the considerable difference between the predicted power of the commercial loads and their actual consumption during flight, more power management techniques are being studied, in order to reduce the wiring weight of the cabin and cargo distribution [31]. The concept is to allow a certain amount of over-installation on the feeders in the secondary cabin and cargo distribution. This means that the total specified current of the allocated loads is larger than the actual rating of the protective device. Eventually, during the flight, the feeders will be subject to overloads. If no power management is involved, this would lead to a trip of the respective protective device (RCCB) and consequently the loss of all functions connected to it. With power management, loads such as the laptop supplies or lighting systems, can be used for power reduction either by simply shedding them or by dimming them in such a way that neither passengers nor cabin crew notice. This is illustrated in Figure 2.8. The diagram on the left shows the effect on the loads when the functionality of a cable is lost due to an electrical fault. In terms of power supply, the result is similar when there is a trip of the RCCB due to overload. In both of them, the supply to all loads is compromised. The rightmost diagram, depicts the principle of power management. By disconnecting load L_2 (assumed to have a power consumption greater than the overload), the problem is solved and the remaining loads are maintained active.

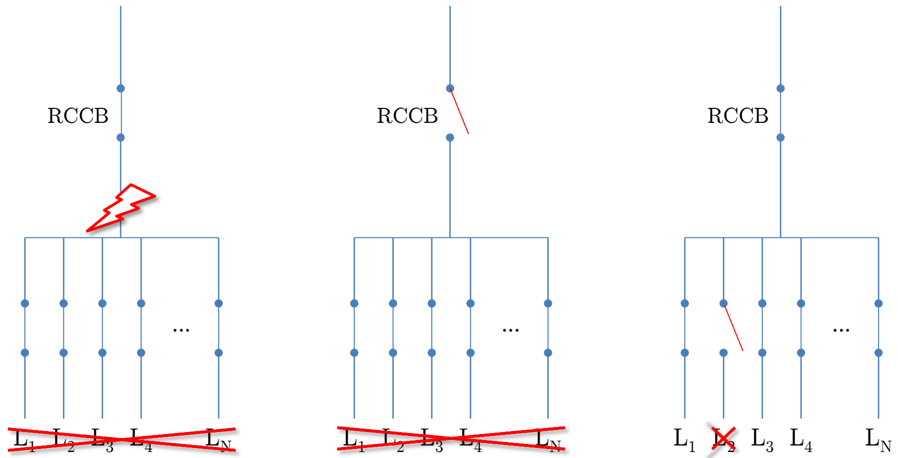


Figure 2.8: Consequences to the electrical supply depending on the system's response to overload. Starting from the left: failure on the cable; RCCB trip; load shedding by PM (adapted from [31]).

2.2 SYSTEM TO BE OPTIMIZED

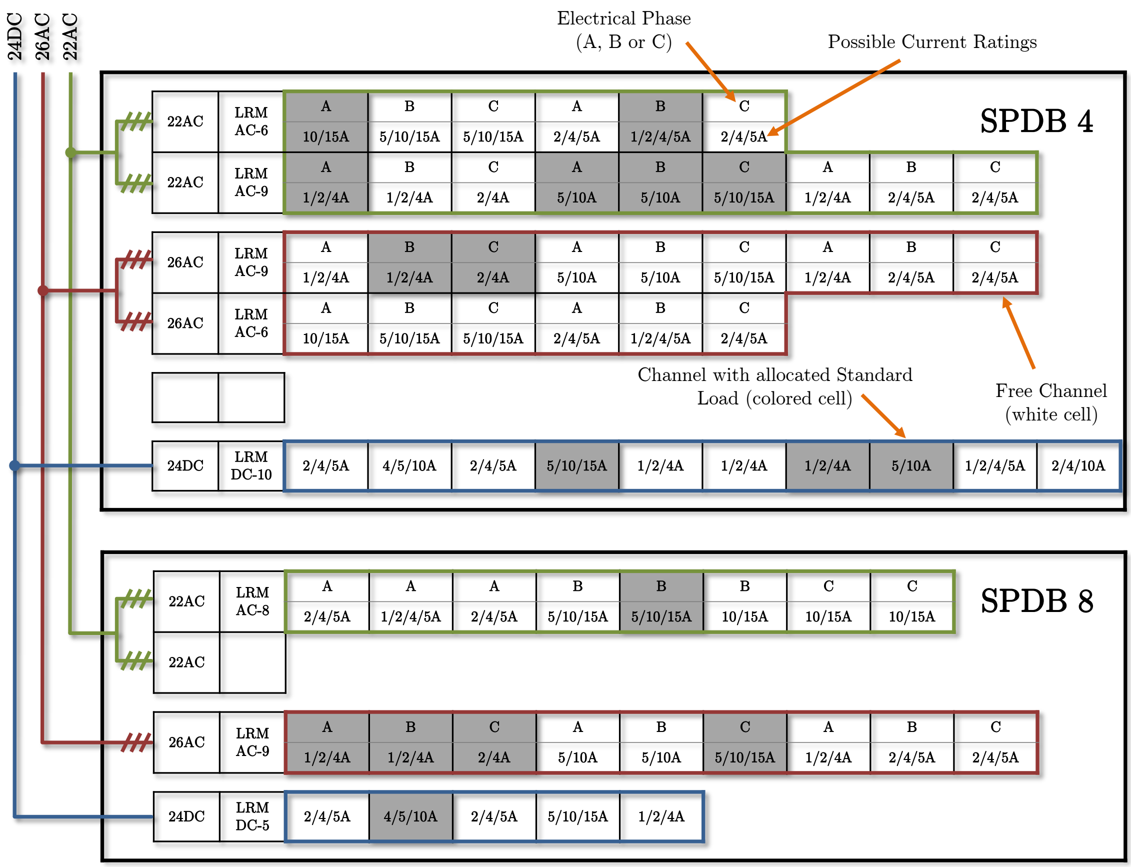


Figure 2.9: Example of power distribution with two SPDBs and three feeders [3], [18], [31], [33], [36].

After having a general idea of a modern aircraft's electrical power system, we can concentrate on the subsystem to be optimized, the secondary power distribution system. As previously mentioned, some of the cabin and cargo loads are always installed (standard loads) and have specific allocation schemes. On the other hand, due to the possibility of customization of cabin and cargo by the airline companies, loads can be added and their allocation to the SSPCs (inside the SPDBs) can be chosen (optional loads). The purpose of this work is to optimize the allocation of the optional loads.

To better understand the proposed problem, we can look at one general example. Figure 2.9 depicts two SPDBs (represented by two boxes), each supplied by three cables belonging to three different feeders: two AC feeders (22AC and 26AC) and one DC feeder (24DC). The AC cables are composed by three wires each, corresponding to the three electrical phases. This example is in compliance with the secondary distribution scheme in Figure 2.6. Note that both these schemes are just general examples; they are only in accordance to make it easier to link the different levels of the electrical power distribution.

Each of the SPDBs has a certain number of LRMs, represented by the rows inside the boxes, containing different numbers and types of SSPCs, represented by the cells with the current ratings (and A, B or C in the case of AC SSPCs). Each of the LRMs is supplied by one of the feeders. This follows from the layout depicted in Figure 1.2. The corresponding feeder is written in each row. For example, the first two cards inside SPDB 4 are supplied by the 22AC feeder.

It is considered that the LRMs can differ, in order to respond to the different loads' demands [3]. This is represented in the example by the type of LRM: "AC-6", "AC-9", "DC-10", etc. Note that they do not only differ in terms of number of SSPCs, but also in their type, as previously indicated.

The colored SSPCs represent the standard loads, i.e., the loads whose position cannot be changed. Looking at the first SSPC (first box, first row), we observe that a load with current rating either 10 or 15 A is already allocated there. This means that no optional load can be connected to this SSPC. Remember that each SSPC cannot supply more than one load (a group of devices supplied together is also considered as a single load). On the other hand, the second SSPC (same box and row) is free, so an optional AC load with either 5, 10 or 15 A of current rating can be connected there.

From now on, in order to simplify the problem analysis for the reader, SPDB will be referred to as box, LRM as card and SSPC as channel.

2.2.1 BOX, CARD AND CHANNEL

Load ID	Optional	AC/DC	Phase	Rating (A)	SPDB (Box)	LRM (Card)	SSPC (Channel)	P_{nom} (W or VA)	$u_{\max}(1)$	$u_{\text{op}}(1)$	...	Permanent	Sheddable
1	Yes	AC	3	10	4	-	-	900	0.8	0.6	...	Yes	Yes
2	Yes	AC	1	2	8	-	-	50	0.5	0.5	...	Yes	No
2	Yes	AC	1	2	8	-	-	150	0.2	0.0	...	No	No
3	Yes	DC	-	4	4	-	-	110	1.0	0.5	...	Yes	No
4	No	AC	3	5	4	2	4-5-6	360	0.9	0.7	...	Yes	No
							⋮						

Table 2.1: Load data example

Table 2.1 shows a simplified example of possible information for allocation purposes. The reader is probably assuming that there is a typing error in the number of the loads, since there are two loads with the same Load ID. Although it looks like that, there is a good reason for this, which will be explained in Section 2.2.5. As we can see in this table, each optional load has a predefined current rating. As previously stated, the optimization of the wiring downstream the SSPCs is out of the scope of this thesis. Therefore, it will be assumed that there is a set of all the available SSPC rating values and the choice of the rating for each load is priorly done based on some criterion (for example, the lower rating possible, in order to minimize the size of the downstream cable). Hence, each load can only be connected to a channel that supplies this value of current rating. Moreover, it will be considered that each optional load already has a specified box (SPDB), which means that we know *a priori* which box the load must be connected to. Does this mean that we can look at each box separately? Not at all. As it will be discussed in detail later, we want to optimize on the feeder level, therefore, the allocation in both boxes will have an impact on the power supplied by the feeder, and consequently an impact on the optimization. Moreover, since there is no specified card, the loads can be connected to any of the cards belonging to the specified box (they must conform to the same type of power, AC loads to AC cards and DC loads to DC cards). This implies that the optimization cannot be held for each feeder independently. But not everything is bad news: since AC channels can only supply AC loads, and DC channels can only supply DC loads, the optimization of the AC power system can be done separately from the DC power system. Unfortunately, as we will see in a moment, this is not completely true!

In order to extend the customization (allocation possibilities), two special cases can be considered. The first relates to optional cards. Looking at the second box (SPDB 8), feeder 22AC supplies two cards. The first card is of the "AC-8" type, but the second card has no information. This is referred as an empty card slot and represents the fact that no card is installed in this position, and so we have the possibility to connect any type of card in this position (we can always leave it free, if this leads to a better solution). Additionally, if no standard loads are connected to a specific card (e.g., fourth card of SPDB 4), the latter can also be considered as changeable.

The second special case is presented in the first box (SPDB 4). Here, there is also a card without information, but additionally no feeder is supplying it. This represents the possibility of changing the cable scheme. In theory, it could be possible to choose which feeder supplies this card. For example, the contiguous cable segment from 26AC feeder or the one from 24DC could be used to supply it. This is precisely the case where AC and DC optimization cannot be done separately. This second special case will not be part of this thesis. It will be assumed that all the connections are priorly defined and only minor changes, like wire thickness, can be performed. As a result the optimization can effectively be done separately for AC and DC.

2.2.2 1-phase vs. 3-phase loads

The column "Phase" in Table 2.1 identifies the type of AC load: single-phase (1) or three-phase (3). Single-phase AC loads can be connected to any of the supplied electrical phases.

Three-phase loads can be regarded as three equal single-phase loads. These three loads consume one third of the total power of the three-phase load and each of them must be connected to a different electrical phase (A, B or C). All the parameters of the three-phase load (except power consumption) are preserved.

The AC cards presented in the example above have different electrical phase distributions among the channels. For example, "AC-8" card has each electrical phase in consecutive channels, while "AC-9" supplies the three electrical phases in groups of three consecutive channels. The latter can be necessary if, for instance, three-phase loads have some predefined connector to the channels. To deal with this difference, cases will be considered where the three resulting loads must be connected to three consecutive channels, starting on a channel supplying phase A.

Example. Consider that the first optional load of Table 2.1 needs to be allocated with a connector described above. The three resulting single-phase loads can only be allocated to three consecutive channels with 10 A rating (starting on phase A), inside the first box. Looking at Figure 2.9, this three-phase load can be allocated, for example, to channels 4, 5 and 6 of the third card inside the first box.

Load	Optional	AC/DC	Phase	Rating (A)	SPDB (Box)	LRM (Card)	SSPC (Channel)	P_{nom} (W or VA)	$u_{\text{max}}(1)$	$u_{\text{op}}(1)$	...	Permanent	Sheddable
1.1	Yes	AC	A	10	4	-	-	300	0.8	0.6	...	Yes	Yes
1.2	Yes	AC	B	10	4	-	-	300	0.8	0.6	...	Yes	Yes
1.3	Yes	AC	C	10	4	-	-	300	0.8	0.6	...	Yes	Yes
2	Yes	AC	Any	2	8	-	-	50	0.5	0.5	...	Yes	No
2	Yes	AC	Any	2	8	-	-	150	0.2	0.0	...	No	No
3	Yes	DC	-	4	4	-	-	110	1.0	0.5	...	Yes	No
4.1	No	AC	A	5	4	2	7	120	0.9	0.7	...	Yes	No
4.2	No	AC	B	5	4	2	8	120	0.9	0.7	...	Yes	No
4.3	No	AC	C	5	4	2	9	120	0.9	0.7	...	Yes	No

⋮

Table 2.2: Load data example with three-phase loads decomposed

Table 2.2 shows the resulting information for optimization input. Notice that the first load from Table 2.1 is now described by the first three loads. These loads have a P_{nom} of 300 VA (one third of the original load) and all the other load's parameters are maintained. The same is done for the standard load priorly defined by number four (loads 4.1, 4.2 and 4.3 in Table 2.2). The column "Phase" is now used to express the need to allocate these three resulting loads to different electrical phases. Load 2, being a single-phase load, can be allocated to any electrical phase.

2.2.3 Power - operational and maximum

The power is probably the most important aspect of the optimization. In the previous tables, a power value P_{nom} is given. It stands for *nominal power* and represents the specified power taken up under nominal voltage $U_{\text{nom}} = 115$ VAC for AC loads and $U_{\text{nom}} = 28$ VDC for DC loads [31]. This value is used for the wiring dimension between the load and the SSPC, as well as the protective current rating. Therefore, it can be regarded as the highest power the load can stand without being damaged. Note again that this value can correspond to a group of equipment that is going to be supplied by one channel, but for our analysis this is regarded as a single load.

Maximum power is defined as the most probable power consumption under the most unfavorable conditions. These are related to external temperature, weather, and other factors, such as the passenger payload [31]. We do not need to go into detail for what it is considered as most unfavorable conditions.

Operational power is defined as the most probable power consumption in normal operating conditions. These represent the common circumstances for most of the flights around the world [32].

Thus, the power values of nominal power P_{nom}, maximum power P_{max} and operational power P_{op} obey the following relation:

$$P_{\text{nom}} \geq P_{\text{max}}(\gamma) \geq P_{\text{op}}(\gamma), \quad \forall\gamma \tag{2.12}$$

where γ is the flight phase (explained below).

2.2.4 FLIGHT PHASES

The *operational power* and the *maximum power* consumptions are flight phase dependent [31]. Flight phase refers to a specific period during the operation of an aircraft. "Taxi", "Take off", "Climb", "Cruise" or "Landing" are examples of flight phases [31].

Since both powers mentioned above are smaller than the *nominal power*, they can be represented as a fraction of the latter, for each of the flight phases. Thus, *operational power* and *maximum power* are calculated by multiplying the *nominal power* by a consumption factor (0-1) for each type of power.

The maximum power for a given flight phase γ, is then given by:

$$P_{\text{max}}(\gamma) = P_{\text{nom}} \times u_{\text{max}}(\gamma), \tag{2.13}$$

where $u_{\text{max}}(\gamma)$ is the consumption factor for maximum power on flight phase γ.

Similarly, the operational power for flight phase γ, is calculated by:

$$P_{\text{op}}(\gamma) = P_{\text{nom}} \times u_{\text{op}}(\gamma), \tag{2.14}$$

where $u_{\text{op}}(\gamma)$ is the consumption factor for operational power on flight phase γ.

Example. The first load (Load 1.1) in Table 2.2, on the first flight phase, has a consumption factor for maximum power and operational power of 0.8 and 0.6, respectively. Since the nominal power is 300 VA, the resulting maximum and operational power, on the first flight phase, are calculated by:

$$P_{\text{max}}(1) = P_{\text{nom}} \times u_{\text{max}}(1) = 300 \times 0.8 = 240 \text{ VA} \tag{2.15}$$

$$P_{\text{op}}(1) = P_{\text{nom}} \times u_{\text{op}}(1) = 300 \times 0.6 = 180 \text{ VA} \tag{2.16}$$

2.2.5 PERMANENT VS. INTERMITTENT

Loads can work in a permanent and/or intermittent way. This is defined based on the usage time. Intermittent operation is related to short term usage and has a duration not larger than 300 sec for AC loads and 30 sec for DC loads [31]. An example of permanent consumption can be an LCD displaying a movie, while intermittent power consumption happens when, for example, a small motor is activated by the passenger to lean its seat. These two operation types and subsequent power consumption are characterized by a *permanent power* and an *intermittent power*, respectively. Hence, loads can have permanent and/or intermittent power values. For each of these situations, a set of values for nominal power and usage fractions is given. In the table, the different type of operation is indicated in the column "Permanent".

Example. Load 2 in Table 2.2 has a permanent nominal power of 50 VA and for the first flight phase has consumption factors of 0.5 for both maximum and operational power. This load also has an intermittent nominal power of 150 VA and consumption factors of 0.2 and 0.0.

This distinction is very important for the verification of the power consumption limits for each cable.

2.2.6 SHEDDABLE VS. NON-SHEDDABLE

As it was already pointed out, there can be situations where over-installation in the feeders is allowed. This happens if power management is used. But for this to happen there must be, among the loads, some that can be shed or dimmed [31]. We will consider the cases of loads that can be shed, distinguishing between sheddable and non-sheddable loads. Hence, in case of effective overload during flight, channels supplying sheddable loads can be disconnected by some kind of power management.

This shedding is based on some predefined priority [20], [28], [30]. Thus, loads with high shedding priority are disconnected first. If this is not sufficient to clear the overload, loads with progressively lower priority are shed.

The distinction between sheddable and non-sheddable is important for power consumption limits calculations.

2.2.7 APPLICABLE LIMITS

A distinction has to be made between power limits and applicable limits. In this work, power limits are defined as the highest power a device (cable, protective device, etc.) can stand before being damaged. With respect to cables, this is directly calculated by multiplying the cable current rating by the operation voltage. Applicable limits are the constraint limits associated with the devices, which for various reasons (power management, type of load operation, etc.), may be different from the power limits.

The actual terminal voltage across the loads can vary due to fluctuations or voltage drops across the lines. So it is usual to consider the limits for the total current of the allocated loads not exactly equal to the limits of the cables and the protective devices. For example, considering a 15% possible variation in the voltage across the terminals of the loads leads to an applicable limit of 87% of the actual power limit of the system [31].

When no power management is considered, the distribution shall not lead to currents above the nominal feeders' ratings in steady state conditions, even when all loads are operating at their maximum power. But inrush currents may exceed this rating for a short time [31]. This indicates that applicable limits may vary in terms of the type of operation – permanent or intermittent.

When power management is available, over-installation can be done. In [31], over-installation factors such as 2 are considered, which means that the total specified current of the loads can go up to twice the actual power limit of the cable or protective device. However, after deactivating all sheddable loads, the normal applicable limits shall be met (for example, the 87% applicable limit described above). As a result, applicable limits may also vary depending on the types of loads, sheddable or non-sheddable. Moreover, in our example in Figure 2.9, consider that, for instance, feeder 22AC has PM, while feeder 26AC has no PM. If an AC sheddable load must be connected, say, to SPDB4, it will be supplied either by 22AC or 26AC, depending on the channel. If supplied by 22AC, it can be included in the over-installation applicable limit. However, if supplied by 26AC, since this feeder has no PM, this load cannot be shed, and therefore it must be considered has a non-sheddable load.

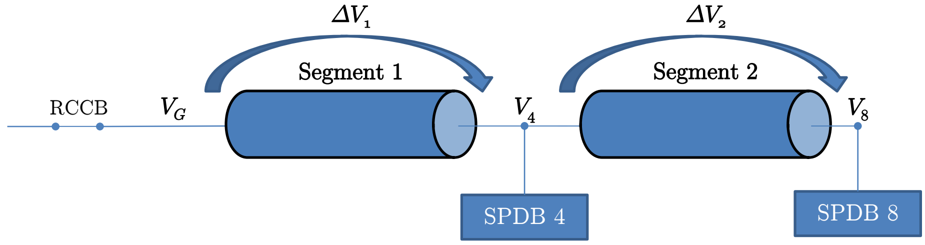
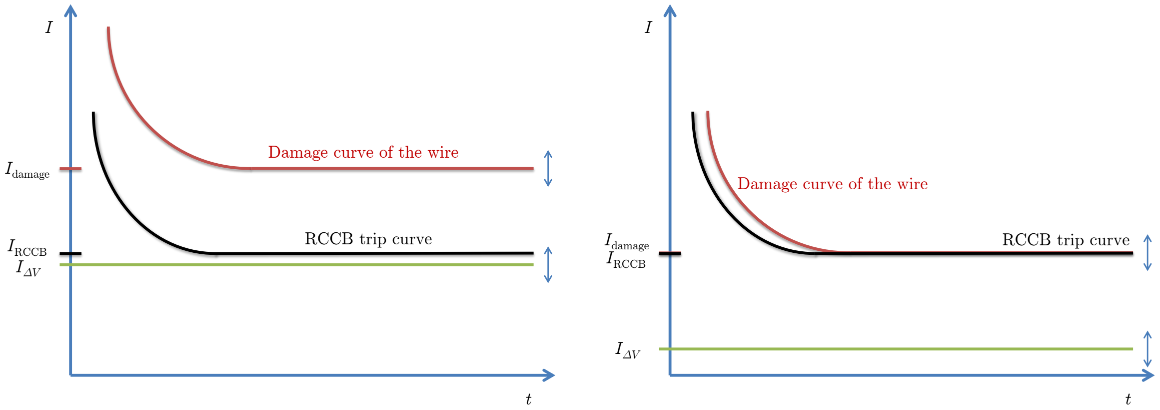


Figure 2.10: Cable segments considered on each feeder (22AC).

Weight optimization will be one of the targets considered in this thesis. In normal conditions, the selection of the protective device rating defines the thinnest wire allowed [31]. If cables with lower rating were used, the RCCB would be unable to protect them.

Looking at Figure 2.10, two main segments are identified for the feeder 22AC. Both segments should have a rating equal or higher than the rating of the protective device, due to the reason discussed above. Cables with higher rating normally have a larger cross-section and are consequently heavier. This means that there is no reason (regarding weight) to prefer a cable with higher rating, if not strictly necessary. But regarding this choice, another aspect must be considered: the voltage drop over the segments. This is represented in the figure by ΔV_1 and ΔV_2, for the first and second segments respectively. As a result, the voltages supplied to the loads will be lower than the system's voltage (in the figure, $V_G > V_4 > V_8$). In order to ensure the correct operation of the loads, these voltage drops must be kept below specified limits [31].



(a) (b)

Figure 2.11: Possible solutions to reduce the voltage drop. Part (a) illustrates a cable with increased cross-section (and consequent higher damage current of the wire), in order to enable a current close to the RCCB limit. Part (b) shows the decrease of the allowed current in the cable ($I_{\Delta V}$) and consequent reduction of its cross- section.

The calculation of the voltage drop over a cable can be done, in a simplified manner, by the following expression (neglecting capacitive and inductive components):

$$\Delta V = l \times r \times I, \tag{2.17}$$

where l is the length of the cable, r is the resistance per unit of length and I is the current. Inspecting the equation, and since changing the length of the cables is not an option, there are two possible ways to reduce the voltage drop: (1) use a cable with lower resistance or (2) reduce the supplied current. The first can be achieved by choosing a cable with higher cross-section, or equivalently, a cable with higher current rating. This measure can be sufficient to enable the maximum current supply (rating of the protective device), but also implies the use of heavier conductors. This is depicted in Figure 2.11a. The maximum allowed current $I_{\Delta V}$ is close to the RCCB rating, by selecting a cable with larger cross-section, expressed by the higher I_{damage}. This value represents the maximum allowed current before damaging the wire. It is also noticeable in this figure that current overloads are allowed without damaging the wire or tripping the RCCB, provided that they are rapidly solved.

On the other hand, the second method can be used to reduce the weight of the cables. Here, instead of allowing a load consumption up to the limit of the protective device, a cable with lower damage current is used (still equal or greater than the RCCB rating), on the condition that the current in the cable does not exceed the value necessary to fulfill the voltage drop constraint. This is illustrated in Figure 2.11b. The damage curve of the wire is close to the RCCB limit, causing a lower allowed current in the cable ($I_{\Delta V}$). This results in a wire with diminished cross-section, leading to weight savings.

Note that the second option (although possible) can be more complicated to implement on the first segment, since all the current needed to supply the loads connected to SPDB 8 must also go through this segment. But it is theoretically possible to reduce the second segment's rating if, for example, the current consumption (and consequent power consumption) of the loads supplied by SPDB 8 are always below a given value, lower than a specified $I_{\Delta V}$. This is a possible solution when it comes to weight reduction.

A problem now arises regarding power management. The current can go over this limit value, but still be below the RCCB's tripping rating. This leads to a malfunction the power management system is not able to detect, since it is not formally an overload. So even if there were sheddable loads supplied by SPDB 8, they would not be shut by the power management system. So, for this case, all loads supplied by SPDB 8 must be considered as non-sheddable when power consumptions are calculated.

In summary, it will be considered that there is a different set of applicable limits for each of the cable segments, which differ according to the following situations:

- feeders with PM,
- feeders without PM,
- segments with PM with allowed current lower than the corresponding RCCB's rating.

Each of the limits can address maximum or operational power and must be verified in all flight phases, since the power consumption varies during flight. As discussed, the limits can contain different combinations of the types of operation and the types of loads: permanent, intermittent, sheddable and non-sheddable.

Load ID	Optional	AC/DC	Phase	Rating (A)	SPDB (Box)	LRM (Card)	SSPC (Channel)	P_{nom} (W or VA)	$u_{\max}(1)$	$u_{\text{op}}(1)$	...	Permanent	Sheddable
1	Yes	AC	3	10	4	-	-	900	0.8	0.6	...	Yes	Yes
2	Yes	AC	1	2	8	-	-	50	0.5	0.5	...	Yes	No
2	Yes	AC	1	2	8	-	-	150	0.2	0.0	...	No	No

Table 2.3: Load data for applicable limits example.

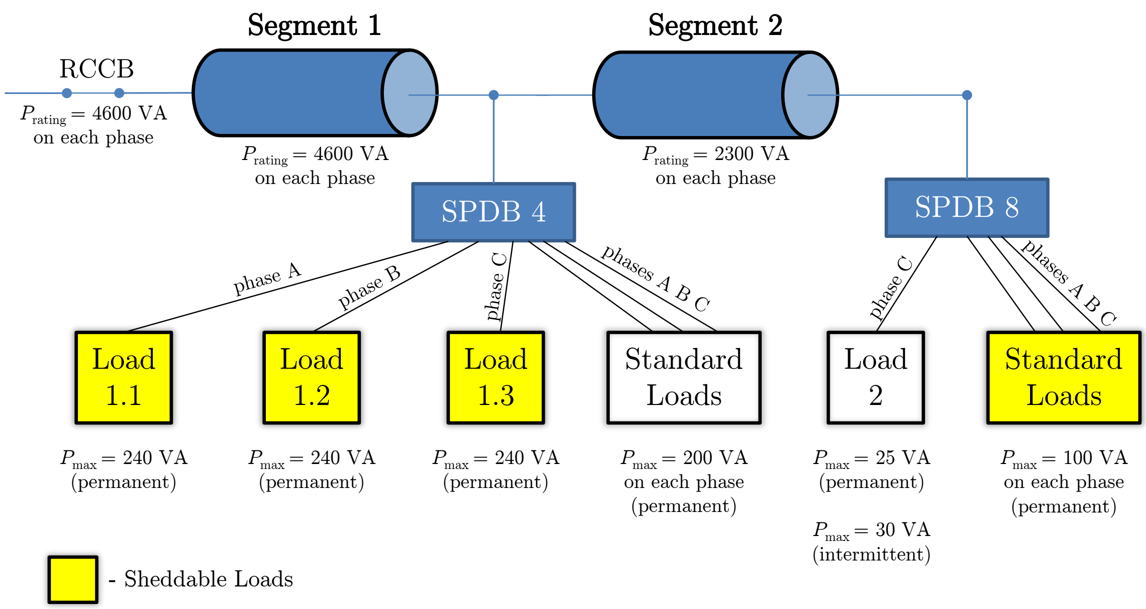


Figure 2.12: Example of the calculation of the applicable limits for an allocation.

ΔV Constraint	Applicable Limit	Maximum Power	Operational Power	Permanent	Intermittent	Non-Sheddable	Sheddable
-	87%	X	-	X	X	X	-
X	87%	X	-	X	X	X	X
-	200%	X	-	X	X	X	X

Table 2.4: Applicable Limits for the example.

Example. Consider now that Loads 1 and 2 from Table 2.3 are connected to cards supplied by feeder 22AC, equipped with power management. Specifically, Load 2 is connected to electrical phase C in the second box (SPDB 8). The three phases of the 3-phase load (Load 1) must be connected to different electrical phases inside the first box (SPDB 4) so, for that particular load, let us consider that the nominal power on each phase is $900/3 = 300$ VA. The corresponding maximum powers can be calculated using (2.13). The example is illustrated in Figure 2.12.

Due to some standard loads, there are already 200 VA of maximum power on each electrical phase allocated to the first box and 100 VA of maximum power allocated to the second box. The standard loads allocated to the first box are non-sheddable, while the ones allocated to the second box are sheddable. Note that the 200 VA go only through the first segment, while the 100 VA go through both segments.

Consider also that the current rating for the RCCB is 40 A (power limit of $115 \times 40 = 4600$ VA on each electrical phase). The first segment has a cross-section that permits a total current of 40 A (while fulfilling the voltage drop limit), so the power limit is considered to be the same (4600 VA). The second segment was chosen on the condition that the total current cannot go over $I_{\Delta V} = 20\ A$ (power limit of $115 \times 20 = 2300$ VA), in order to fulfill the voltage drop limit.

For the RCCB and for the first cable segment, assume the maximum power for permanent operation cannot go over 87% of the corresponding power limits and that an over-installation with factor 2 is allowed. All types of loads and all types of operation are considered. This is summarized in Table 2.4, on the first and third lines. Note that maximum powers of the optional loads were calculated using the consumption factor $u_{\max}(1)$ (for the first flight phase). The applicable limits for maximum power on the first flight phase are given by:

RCCB and first segment:
Phase A

$$\begin{aligned} 200 &\leq 0.87 \times 4600 \\ 200 + 100 + 240 = 540 &\leq 2 \times 4600 \end{aligned}$$

Phase B

$$\begin{aligned} 200 &\leq 0.87 \times 4600 \\ 200 + 100 + 240 = 540 &\leq 2 \times 4600 \end{aligned}$$

Phase C

$$\begin{aligned} 200 + 25 + 30 = 255 &\leq 0.87 \times 4600 \\ 200 + 100 + 240 + 25 + 30 = 595 &\leq 2 \times 4600 \end{aligned}$$

As previously explained, since the maximum current allowed in the second segment is below the RCCB's rating, over-installation is not allowed and all loads should be considered as non-sheddable. This is summarized in the second line of Table 2.4. The applicable limits are calculated by:

second segment:
Phase A

$$100 \leq 0.87 \times 2300$$

Phase B

$$100 \leq 0.87 \times 2300$$

Phase C

$$100 + 25 + 30 = 155 \leq 0.87 \times 2300$$

Finally, observe that this is a calculation for a single flight phase. All the applicable limits must be calculated for all flight phases.

CHAPTER 3

Optimization

After analyzing the system in detail, it is now time to find a way to optimize the allocation of the loads. The broad field of study covering the topic of optimization is known as Operations Research (OR) and can be defined as a scientific approach to decision-making. It aims to find the best way of operating a system, usually involving the allocation of scarce resources [38]. This is closely related to the concept of mathematical modeling, which is the process of describing pertinent features of a problem by means of a collection of variables and their relationships. Therefore, OR can be regarded as the study of how to form a mathematical model of a complex problem and its analysis to reach possible solutions [29].

3.1 History

Due to the exceptional growth of organizations in size and complexity, following the industrial revolution, there was a tremendous increase in the division of labor and distribution of responsibilities. Although substantially improving companies' productivity, this specialization raised new problems. One of these is the tendency of different divisions to have their own activities and goals, thereby sometimes losing sight of their impact in the organization's overall objectives and productivity. What is best for one organization's component can be detrimental to another. A second relevant problem arising from this division of work is the difficulty to allocate the available resources to the different activities in the most effective way for the organization as a whole. The need to find solutions for this kind of problems led to the development of OR [16].

Despite earlier attempts to apply scientific methods to the management of organizations and other decision-making problems, the beginning of operations research as a formal discipline has been attributed to the military services in World War II. During this period, the need to allocate scarce resources to various military operations in an effective manner led the U.K. and then the U.S. military management to ask a large number of scientists to apply scientific approaches to solve these and other types of problems. They were asked to do *research on* (military) *operations* [16]. As an example, due to the research on bombing raids, there was at least a 1000 percent increase in bombs on target [13].

After the successful application (in a mathematical point of view) of OR in war context, the idea of applying the same principles to solve a broader type of problems was logical. Consequently, scientists felt motivated to pursue relevant research in this field, and hence important enhancements resulted. This contributed to the rapid growth of OR in the decades following the war. Many tools, such as linear programming, were greatly enhanced during this period.

The next boost came with the development of computers over the following decades. This brought huge improvements in dealing with the large amount of computation usually required in problems of OR.

3.2 PHASES OF OPERATIONS RESEARCH

The process of an OR study can be described by the following phases [16]:

Define the problem and gather relevant data
Defining the problem involves specifying the appropriate objectives and the parts of the problem that must be studied before a model can be developed. There should also be considerable time devoted to gathering data. This is needed to gain a good understanding of the problem and also to identify inputs for the mathematical model, such as conditions that shall be met for an appropriate solution.

Formulate a mathematical model for the problem
After understanding the problem to be optimized, a collection of variables defining the system must be developed. The mathematical model must also contain the relations between these variables, as well as the way they affect the objectives of optimization. Due to the usually high complexity of the problem, it is easier to start with a model for a simplified version of the system, and then gradually add more information until the total system is modeled.

Develop a computer-based procedure for deriving solutions using the model
This step usually involves the use of priorly defined standard algorithms and can be performed with the help of available software.

Test the model and refine it as needed
The first version of a complex mathematical model will most definitely contain many flaws (analogous to the inevitable bugs in the first version of a complex computer program [16]). Therefore tests (preferably with real data) must be done and the errors must be corrected. Keep in mind that, due to the complexity of some systems, some minor flaws may never be detected.

Implement
After the solution procedure is applied to the model, additional computer programs may be needed to implement the results. These additional programs usually aim to make the optimization procedure easier for people using the tool, combining it with previously used software in the organization. Documentation on how to reach the optimal results shall also be written. These phases also take considerable time.

3.3 DEFINITIONS

There are three fundamental concepts in OR models, these are [29]:

Decisions open to decision makers,

Constraints that limit the decisions, and

Objectives making some decisions better than others.

In this work we will look exclusively at mathematical programs, that is, models that represent the decisions as mathematical variables and seek the maximization or minimization of objective functions of the variables. These are subject to limits on possible decisions, i.e., constraints. Let us define some important concepts [16], [29]:

Definition 1. *A <u>Feasible solution</u> is a choice of values for the variables that satisfies all the constraints.*

Definition 2. *An <u>Infeasible solution</u> is a choice of values for the variables that violates at least one constraint.*

Definition 3. *The <u>Feasible region</u> is the collection of all the feasible solutions.*

Definition 4. *The <u>Optimal solution</u> is a feasible solution that achieves objective function values at least as good as any other feasible solution.*

When defining a model, it is also relevant to define the concepts of tractability and validity. *Tractability* is the degree to which the model of the system can be analyzed. *Validity* is the degree to which the inferences drawn from the model apply to the real system. There is always a trade-off between these two concepts. Ideally, we want the system to be as valid as possible, but normally simplifications have to be made so that the analysis can be tractable.

3.4 MIXED-INTEGER LINEAR PROGRAMMING

The development of Linear Programming (LP) is considered to be one of the most important scientific advances in the mid-20th century. LP is a tool for solving optimization problems and, since the development of the simplex algorithm (a remarkably efficient solution procedure developed by George Dantzig, in 1947), it has been extensively used in activities as diverse as banking, education and trucking, saving many companies and businesses millions of dollars [16], [38].

The adjective *linear* means that all mathematical functions involved in this model need to be *linear functions*. Before going any further, we shall recall the definition of linear function:

Definition 5. *A function $f(x_1, x_2, \ldots, x_n)$ of $x_1, x_2, \ldots, x_n$ is a linear function if and only if $f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$, for some set of real numbers $c_1, c_2, \ldots, c_n$.*

Specifically, the single *objective function* should be a linear function of the decision variables. Furthermore, each *constraint* is formulated by requiring a linear function of the decision variables to be either equal to, not less than, or not more than, a scalar value. A standard condition states that each decision variable must be nonnegative [34].

Considering a set of real variables $x_1, x_2, \ldots, x_n$, a linear programming problem takes on the form:

$$\text{minimize or maximize} \quad c_1x_1 + c_2x_2 + \cdots + c_nx_n \tag{3.1}$$

$$\begin{aligned} \text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n (\leq, =, \text{or } \geq) b_1 && \text{(3.1a)} \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n (\leq, =, \text{or } \geq) b_2 && \text{(3.1b)} \\ & \cdots && \text{(3.1c)} \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n (\leq, =, \text{or } \geq) b_m && \text{(3.1d)} \\ & x_j \geq 0, \ \forall j = 1, \ldots, n. && \text{(3.1e)} \end{aligned}$$

Values c_j, for all $j = 1, \ldots, n$ are referred to as objective coefficients, and can be regarded as an importance weight of their corresponding decisions to the objective value. The values $b_1, \ldots, b_m$ are the right-hand side values of the constraints, and often represent amounts of available resources (commonly for $\leq$ constraints) or requirements (commonly for $\geq$ constraints). Typically, the a_{ij} values describe how much resource/requirement i is consumed/satisfied by the decision j [34].

Although at first it can seem very restrictive, we shall see that with the help of some tricks, this type of formulation can be used in a large number of situations, even non-linear functions.

At this point, it is probably a good idea to see how this model could fit our problem. A simplified version (neglecting, for now, some of the more specific requirements) is how to allocate a defined number of loads to available channels, while fulfilling constraints such as power limits. Trying to do an analogy with the LP model described above: the available choices are the positions where the loads are allocated, therefore, our decision variables should somehow reflect the allocation of a load to a channel. This can be achieved, for instance, considering that each variable x_j is associated with a load-channel pair. Then, $x_j = 1$ represents the allocation of the corresponding load to that specific channel, while $x_j = 0$ means otherwise. The available resources are the channels and the power that the feeders can supply. These would apparently fit in $\leq$ constraints, since an allocation of a load to a channel or a feeder consumes these available resources. The coefficients a_{ij} denote the amount of resource consumed with variable j. Hence, in the case of the power supplied by a feeder, these coefficients should represent the amount of power needed for supplying the allocation of the load assigned to variable j. Thinking of the characteristics of the loads, the power consumptions would perfectly fit this concept. In the case of the channels, the amount of resources could be regarded as the number of channels occupied by a specific load allocation.

The more attentive reader may now find a flaw in this analysis. Linear programming assumes the variables $x_1, \ldots, x_n$ to be real-valued. The problem here is that our decision variables of load allocation cannot be real-valued since we cannot allocate, for example, half of a load to one channel and another half to a different one. Loads, as well as channels, are integer variables.

If we have restrictions on the variables to be integer, we get an Integer Linear Programming (IP) problem. Furthermore, if these variables are restricted to take either the value 1 or 0 (normally standing for yes or no decisions), the problem is said to be a Binary Integer Linear Programming (BIP) problem. A more general linear programming model, which contains both IP and BIP, is the Mixed-Integer Linear Programming (MIP) , where some of the variables, but not necessarily all, are integer-valued.

Before going into detail on how to model and solve MIP problems, a graphical solution for a simple Linear Programming (LP) problem will be explained. This type of analysis can only be performed when the problem has two (maybe three) variables. But it will give us a better understanding of LP and it will also be very important when discussing ways to solve MIP problems.

GRAPHICAL ANALYSIS OF AN LP

Consider an LP formulation with two variables and the following constraint (among others):

$$-2x_1 + 6x_2 \leq 15. \tag{3.2}$$

This constraint can be written with respect to x_2:

$$x_2 \leq \frac{5}{2} + \frac{1}{3}x_1. \tag{3.3}$$

Equation $x_2 = \frac{5}{2} + \frac{1}{3}x_1$ can be plotted in the x_1x_2 plane as depicted in Figure 3.1a. Moreover, the colored part bounded by this equation represents the region that satisfies constraint (3.3).

Suppose now that the remaining constraints of this formulation are plotted in the same manner and we end up with the bounded region depicted in Figure 3.1b. The colored region that satisfies all constraints represents the feasible set of the problem, that is, the region which contains all the feasible points of the formulation. Since this region results from the conjunction of all the constraints, and these are based on linear functions, the resulting set is *convex*:

Definition 6. *A set of points S is a convex set if the line segment joining any pair of points in S is wholly contained in S.*

Particularly important are some of the points of constraints' intersections. First, we must define convex combination:

Definition 7. *Given two points y and $z$in $\mathbb{R}^n$, a point x is said to be a convex combination of y and z if there exists a scalar $\alpha \in [0, 1]$ such that $x = \alpha y + (1 - \alpha)z$. Intuitively, a convex combination of two points lies on the line segment that joins them.*

The intersection points of interest are known as vertices:

Definition 8. *Let $S \subseteq \mathbb{R}^n$ be a convex set. We say that a point $x \in S$ is vertex of S if x is not a convex combination of two other points in S.*

When referring to a feasible region of an optimization problem, these points are said to be corner-point feasible (CPF) solutions:

Definition 9. *A corner-point feasible (CPF) solution is a solution that lies at a vertex of the feasible region.*

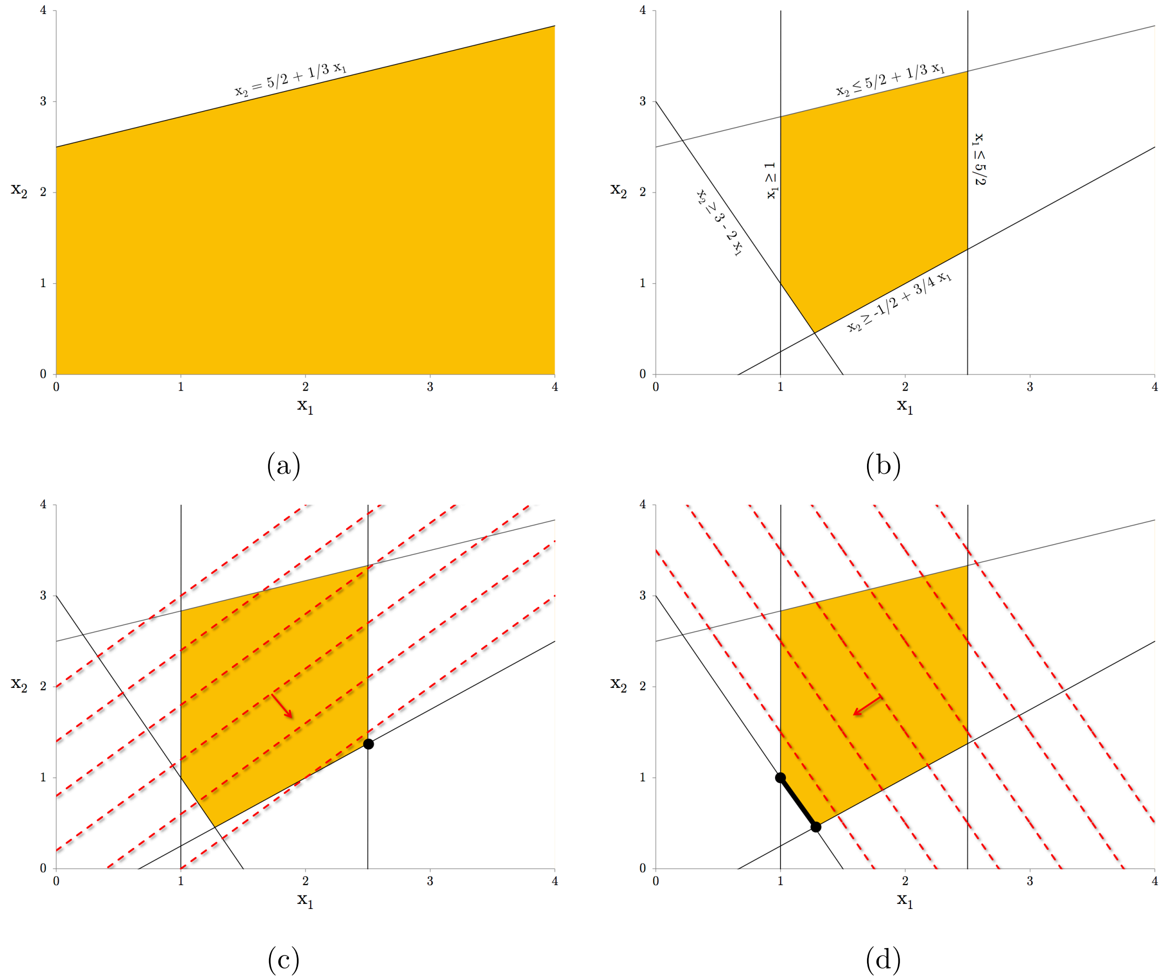


Figure 3.1: LP formulation.

Suppose now that we want to represent the objective function in the same plot as the feasible region. For instance, let us consider that we have the objective function:

$$\text{minimize} \quad -x_1 + x_2. \tag{3.4}$$

Let us pick the point (1,2) and evaluate the result:

$$-(1) + (2) = 1.$$

The value of the objective function for this feasible point is then equal to 1. We can define the points (x_1, x_2) that have objective value equal to 1, i.e.,

$$-x_1 + x_2 = 1, \tag{3.5}$$

or in a more convenient way for plotting,

$$x_2 = 1 + x_1. \tag{3.6}$$

This resulting line is said to be a contour [29]:

Definition 10. *Contours are curves (usually dashed) through points having equal objective function value.*

Note that if we now calculate the contour with objective value 0.5, we get

$$x_2 = 0.5 + x_1, \tag{3.7}$$

which is parallel to (3.6). In fact, all the contours of an LP are parallel between them. These contours can be plotted together with the feasible region, as depicted in Figure 3.1c, and a small arrow perpendicular to the contours shows the direction in which the objective function improves.

It is easy to see in this plot that the marked corner point is the optimal solution for the depicted formulation, since this point is contained in the contour with the best optimal value that intersects the feasible region.

In Figure 3.1d, there is another example with the same feasible region, but a different objective function (notice the difference in the contours). Here, there is an infinite number of alternative optimal solutions, since one of the constraints is parallel to the contours. The following rule can be stated [16]:

Proposition 1. *Consider an LP problem with feasible solutions and a bounded feasible region. If the problem has exactly one optimal solution, it must be a CPF solution. If the problem has multiple optimal solutions, at least two must be CPF solutions.*

This is exactly what can be verified in Figures 3.1c and 3.1d.

3.4.1 MODELING MIP PROBLEMS

Translating a problem description into a formulation of an MIP (the same applies for an LP) should be done systematically, and typically involves three steps [40]:

1. Define a set of decision variables that represent the choices to be optimized.
2. Use these variables to define a set of constraints in such a way that the feasible points correspond to the feasible solutions of the problem.
3. State the objective function using these variables.

It is very common, though, to realize that the initial set of defined variables proves to be inadequate. If this happens, an additional or alternative set of variables must be defined, and another iteration of steps 2 and 3 is done. Thus, this process works as a loop.

When the general formulation of an LP problem was defined, it was stated that it could be applied even to non-linear functions. Here we speak about some particular cases interesting for this work.

Consider a generic *minimax* (minimize a maximum) problem with n variables:

$$\text{minimize} \quad \max_j (c_j x_j) \tag{3.8}$$

$$\text{subject to} \quad x_j \geq 0, \ \forall j = 1, \ldots, n \tag{3.8a}$$

$$x_j \in \mathbb{Z}. \tag{3.8b}$$

The objective function is not linear, but this *minimax* structure can be solved using a small trick. First, we define the variable U as the maximum of all the $c_j x_j$. Then, U is minimized on the condition that it must be at least as large as all the values $c_j x_j$ [34]. The formulation is thus written as

$$\text{minimize} \quad U \tag{3.9}$$

$$\text{subject to} \quad (3.8a) - (3.8b)$$

$$U \geq c_j x_j, \ \forall j \in N. \tag{3.9a}$$

The objective function is now linear, and since its optimal result will be the lowest possible value U can take, the solution is equal to the maximum value among all $c_j x_j$, exactly the result we wanted in the first place. Note that (3.9a) can be written according to the general form priorly defined, that is,

$$c_j x_j - U \leq 0, \ \forall j \in N.$$

Imagine now a situation where the variables must verify only one out of a set of k constraints:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \tag{3.10a}$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \tag{3.10b}$$
$$\cdots$$
$$a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n \leq b_k. \tag{3.10c}$$

This can be achieved with the help of a set of k variables and a sufficiently large constant M:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 + M(1 - z_1) \tag{3.11a}$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 + M(1 - z_2) \tag{3.11b}$$
$$\cdots$$
$$a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n \leq b_k + M(1 - z_k) \tag{3.11c}$$
$$\sum_{i=1}^{k} z_i = 1 \tag{3.11d}$$
$$z_i \in \{0, 1\},\ i = 1, \ldots, k. \tag{3.11e}$$

Notice that according to (3.11d) and (3.11e), exactly one of the variables z_i must be equal to 1 and all the others equal to 0. For all the $z_i = 0$, the right-hand side of the constraints will be $b_i + M$, and since M is a sufficiently large constant, the inequalities have no effect. On the other hand, the unique $z_i = 1$ will make the second term vanish, and the right-hand side of the corresponding constraint will be the initial value b_i. We successfully selected only one of the constraints to be verified. This can be extended to situations where, for example, at least l out of k constraints must be met, by defining (3.11d) as: $\sum_{i=1}^{k} z_i \geq l$. These extra variables z_i and the constant M can also be used to formulate dependencies between the constraints (known as either-or constraints [16]):

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 + Mz \tag{3.12a}$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 + M(1 - z) \tag{3.12b}$$

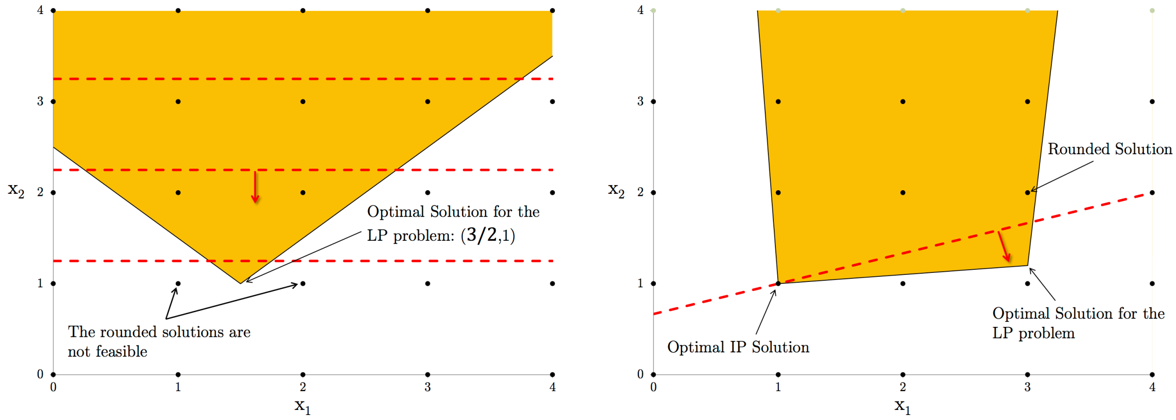
$$z \in \{0, 1\}. \tag{3.12c}$$

So, if $z = 0$, only the first constraint remains active. If $z = 1$, only the second constraint has to be verified. As we will see later, the choice of the value of M can have an impact on the search for the optimal solution.

A few more good examples, such as multiplication of variables and if-then constraints, as well as tricks to solve them, can be found in [29], [34].

3.4.2 SOLVING MIP PROBLEMS

It was stated before that extremely efficient algorithms to solve LP problems exist. One of the first ideas that could come to mind would be to solve the LP problem and then round the solution to the nearest integer. This is often insufficient, as the following examples are going to show. First let us look at Figure 3.2a. Here, after finding the optimal solution for the LP problem, the rounding procedure would select infeasible points, hence the MIP problem would not be solved. But even if we have a procedure to always select the closest feasible point, it is not guaranteed that the optimal solution is found. Looking now to Figure 3.2b, the closest point to the LP solution is far away from the optimal MIP solution.



(a) (b)

Figure 3.2: Rounded LP solution vs. MIP solution.

But if we have only discrete decisions (integer solutions), why not enumerate them and try to find the one with the best possible value? This idea may seem appealing. The article [34] has an interesting example related to this method (that is adapted here): consider a problem with n binary variables, and suppose that each configuration of the variables can be tested for feasibility and scored (according to the objective function) with n computer operations. Since each variable is binary, this leads to $2^n \cdot n$ operations. If a problem with $n = 55$ were solved using a computer able to process 1,000 trillion operations per second (at the time of writing, only 37 supercomputers of the world's top 500 can do so), it would take approximately half an hour to the job. It could still look tempting to let the (super)computer run for a few hours in case of a more difficult problem. However, for $n = 60$, the same computer would require almost 20 days to finish the job, for $n = 70$ it would take 2.5 years, and for $n = 80$ we would wait 3067 years. The computational growth rate for this type of problems is impressive, but even if a great technological leap led to personal computers 100 times faster (100,000 tetraflops) than this supercomputer, they would still only be able to solve problems with $n = 65$ within 17 hours. Enumeration of a $n = 100$ problem (a discrete model with 100 variables is not particularly large) would take approximately 402,000 centuries. As [29] says: "Too long for the most patient of decision-makers to wait". Although impressive, computer speedups are no match for exponential enumeration problems [34].

There should now be enough motivation to seek alternative methods for solving this type of problems.

As previously said when explaining the phases of an Operations Research (OR) study, nowadays solving MIP problems is usually done with the help of a software containing already implemented solving methods. In the present case MATLAB® will be used. The function `intlinprog`, inside the Optimization Toolbox version 7.0 (2014a), is responsible for running these methods. The analysis of the algorithms of the solver not only provide a good insight on how to solve MIP problems, but can also provide good clues on how to efficiently formulate mathematical models. The performance of a specific formulation is tightly related to the algorithms used by the solver, therefore this analysis can be helpful when deciding, for example, how to formulate constraints. The analysis will be done considering minimization problems. The same applies to maximization problems, with the appropriate modifications.

Consider the IP problem:

$$
\begin{aligned}
z = \min \quad & c^{\mathrm{T}} x && (3.13)\\
= \min \quad & c_1 x_1 + c_2 x_2 + \ldots + c_n x_n. && (3.14)
\end{aligned}
$$

We can start by thinking of how we can prove that a given point $x^* = (x_1, x_2, \ldots, x_n) \in X \subseteq \mathbb{Z}^n$ is optimal. The answer can come from finding a lower bound $\underline{z} \leq z$ and an upper bound $\overline{z} \geq z$, such that $\underline{z} = \overline{z} = z$. So, the basic idea to find an optimal solutions is to have an algorithm that finds a decreasing sequence of upper bounds:

$$\overline{z_1} > \overline{z_2} > \ldots > \overline{z_u} \geq \overline{z}, \tag{3.15}$$

and an increasing sequence of lower bounds:

$$\underline{z_1} < \underline{z_2} < \ldots < \underline{z_l} \leq \underline{z}, \tag{3.16}$$

and stop when

$$\overline{z_u} - \underline{z_l} \leq \epsilon, \tag{3.17}$$

where ϵ is some small nonnegative value [40].

Primal Bounds. Note that every feasible solution $x^* \in X$ provides an upper bound (lower bound in case of a max problem). These bounds are also referred to as primal bounds, and finding feasible solutions is basically the only way to obtain them. Depending on the complexity of the IP problem, it can be very easy or extremely hard to find feasible points [38].

Dual Bounds. The lower bounds of a minimization problem are called dual bounds. The most important approach is by "relaxation". The idea is to replace the IP problem by a simpler optimization problem whose optimal value is at least as small as z (lower or equal). There are two somewhat obvious possibilities for this [38]:

- enlarge the set of feasible solutions, or
- replace the min objective function by a function that has the same or a smaller value everywhere.

In the first one, the feasible set from the true model is a subset of the feasible set of the relaxed model. So the optimal solution is still a feasible point of the relaxation.

The second implies that if the relaxed problem is to minimize $f^{\mathrm{T}}x$ and

$$f^{\mathrm{T}}x \leq c^{\mathrm{T}}x,\ \forall x \in X \subseteq \mathbb{Z}^n, \tag{3.18}$$

then, for both types of relaxations, the optimal value z^R must be at least as small as z, i.e., $z^R \leq z$.

It should be now apparent that an optimal value of any relaxation of a minimize model yields a lower bound (dual bound) to the optimization problem.

In MATLAB®, Linear Programming Relaxations are used. Basically, the relaxations are performed by treating any discrete variable as continuous, while retaining all the other constraints. Due to the possibility of further applying the known and efficient techniques for LP problems, these are by far the most used relaxation form [29]. Note that this type of relaxation falls inside the first group: enlarge the set of feasible solutions. Moreover, these LP relaxations have some additional interesting properties, that follow directly from their definition [40]:

- If an LP relaxation is infeasible, so is the full model it relaxes.
- If an optimal solution of an LP relaxation is a feasible point of the full model it relaxes, the solution is also optimal in the latter.

Branch and Bound. We previously excluded the idea of enumerating and testing all possible combinations of integer variables. But what if we could deal with these enumerations in large classes, exploring only those that could contain an optimal solution, and doing so without explicitly enumerating all its elements? In this way, only some of the classes would have to be searched in detail.

This is the basic idea of branch and bound. This method will be explained with one good example taken from [34]. Consider the following MIP formulation (which is also an IP formulation):

$$\text{minimize} \quad 4x_1 + 6x_2 \tag{3.19}$$

$$\text{subject to} \quad 2x_1 + 2x_2 \geq 5 \tag{3.19a}$$

$$x_1 - x_2 \leq 1 \tag{3.19b}$$

$$x_1, x_2 \geq 0 \quad \text{and integer.} \tag{3.19c}$$

Moreover, consider that LP relaxation is applied. The optimal solution of this relaxation is (1.75, 0.75), and 11.5 is the resulting value for the objective function (see Figure 3.3).

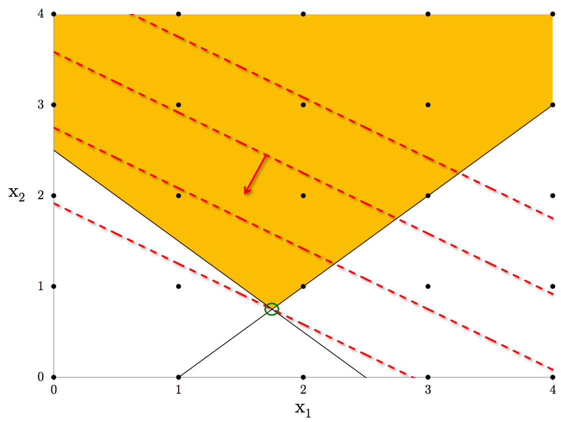
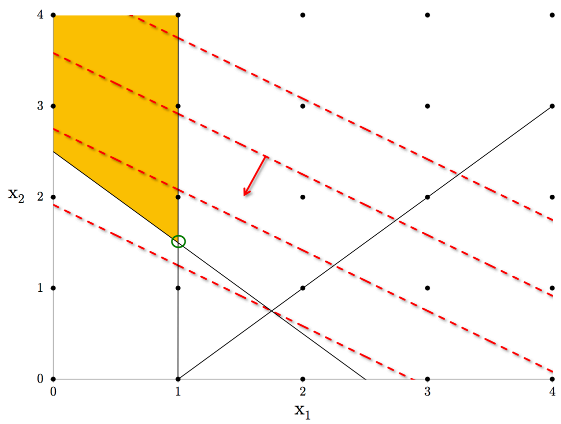
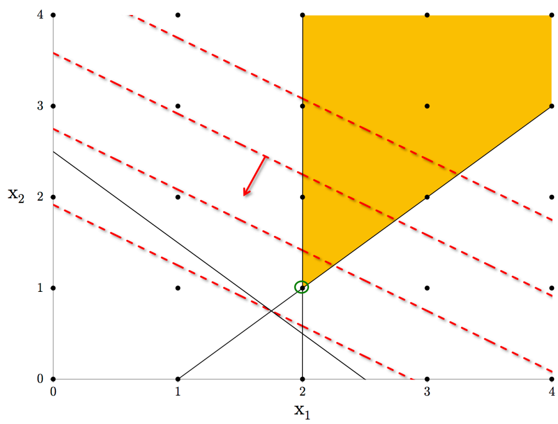


Figure 3.3: LP relaxation (example from [34]).

This value is a lower bound (dual bound), and hence, no value lower than 11.5 can be obtained for the objective function. Unfortunately, this is not an integer solution and, therefore, it is not a feasible point for the MIP problem. But we can split the problem into two subproblems (nodes) by observing that all feasible solutions have either $x_1 \leq 1$ (node 1) or $x_1 \geq 2$ (node 2). With this extra constraint, we end up with the feasible regions of Figures 3.4a and 3.4b, respectively. This process is known as branching, and we could have also branched on x_2. Now, if LP relaxation is applied to both resulting nodes, the optimal solution of the first ($x_1 \leq 1$) is $x_1 = 1$ and $x_2 = 1.5$, with objective value 13. The optimal solution of the second ($x_1 \geq 2$) is $x_1 = 2$ and $x_2 = 1$, with objective value 14. Note that the solution for the second subproblem is an integer solution, hence a feasible point of the MIP problem. There is no need to explore this node further, since we already have an MIP optimal solution for it. This node is said to be *fathomed by integrality*. Solution (2,1) is stored and is called the *incumbent solution*. If no better solution is found, this will be our optimal solution.



(a)



(b)

Figure 3.4: Branch and bound: first branch (example from [34]).

At this point there is one active node ($x_1 \leq 1$). Active means that it must be explored still, since there is a possibility that a better solution than the actual incumbent is contained in this node. The node corresponding to the initial problem is not active anymore, since it was branched, and the node corresponding to ($x_1 \geq 2$) is not active because the best integer solution for that region was already found. Now, the active node is recursively branched, for example, to the nodes with the conditions $x_1 \leq 1 \wedge x_2 \leq 1$ (node 3) and $x_1 \leq 1 \wedge x_2 \geq 2$ (node 4), as depicted in Figures 3.5a and 3.5b, respectively.

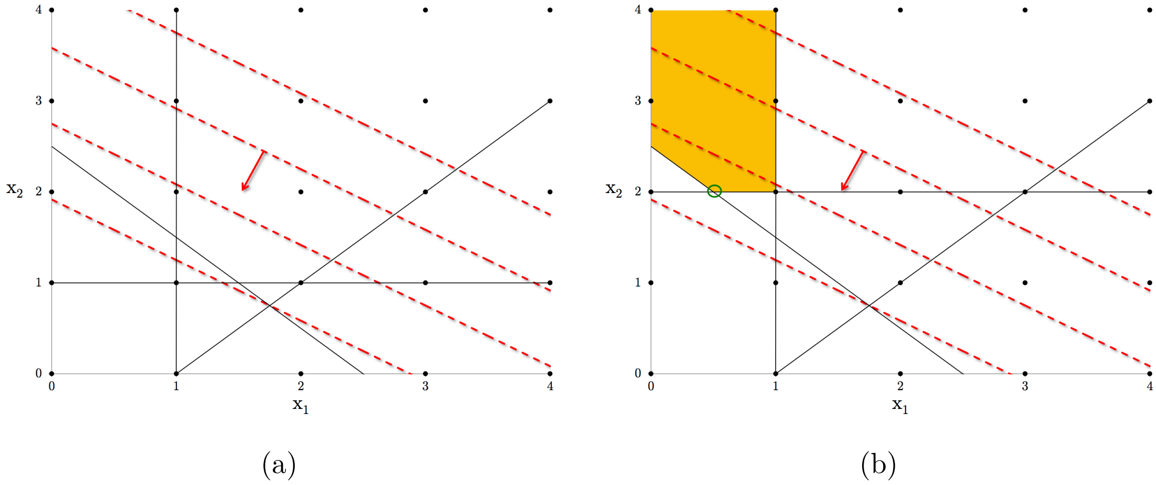


Figure 3.5: Branch and bound: second branch (example from [34]).

Analyzing now these two resulting nodes, node 3 is empty, since no point can satisfy simultaneously all the constraints. So, there is no point in further exploring this node, and it is said to be *fathomed by infeasibility*. The optimal value of LP relaxation for node 4 is the point (0.5, 2) with objective value 14. Although the optimal MIP solution for this node was not found, we know that the value can never be lower than 14 (it is a dual bound for this node). Hence, the analysis of this node will not improve the incumbent solution, and consequently there's no point in exploring it further (we only want to find one optimal solution, even if multiple exist). Node 4 is said to be *fathomed by bound*.

A resulting branch and bound tree, as depicted in Figure 3.6, summarizes the procedure. Note that each node of the tree represents a feasible region.

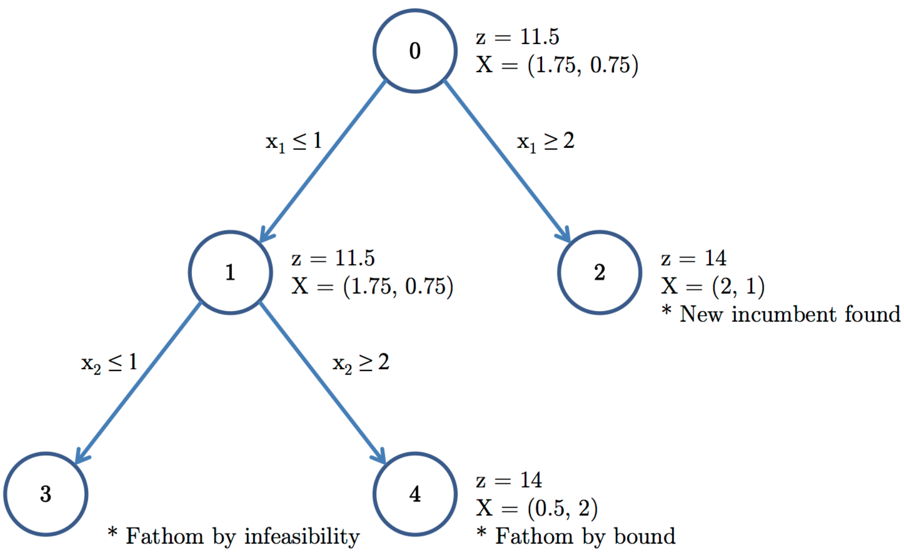


Figure 3.6: Branch and bound tree for the example (example from [34]).

At some point during this procedure, it was decided to branch with respect to variable x_1 first, but there is no obligation to do so. Also, the sequence of analysis of the different nodes was done without any rule. So the question poses itself: how to choose the variables to branch and which rules to follow when deciding the next node to be analyzed? We will not go into detail on this, but there is also research on these types of problems. For example, to decide which variable to branch, a common choice is *the most fractional variable* (variable with fractional value 1/2 is the best) [40]. Other more sophisticated algorithms attempt to choose branching variables that would lead to early fathoming [16].

To choose the node, there are arguments in favor of descending as fast as possible in the branch and bound tree, while others focus on the bounds of the nodes [40].

In this branch and bound, the LP relaxation is clearly a central point. If strong bounds come from solving the different LP problems, more nodes can be fathomed by bound and the performance can improve strongly.

Let us look back to the example of Figure 3.1, where the graphical analysis of an LP formulation was analyzed. Consider now that the same formulation is used to solve an MIP problem (x_1 and x_2 are now integers). The only difference is that the feasible set is not all the region that falls inside the constraints, but only the integer points that reside there. Therefore, there is an infinite number of formulations for this feasible integer set. In Figure 3.7, the initial formulation P_1 and two more

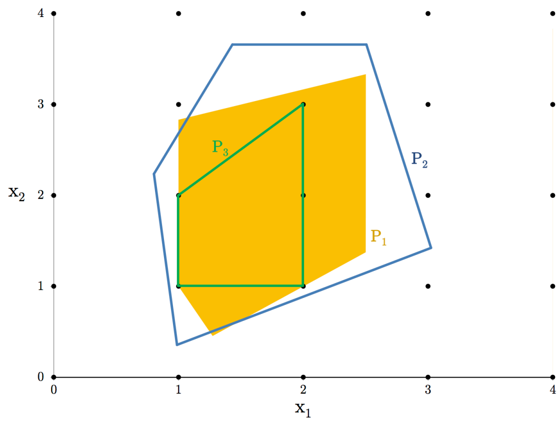


Figure 3.7: Different formulations.

(P_2 and P_3) are presented. Note that all of them contain the same integer points. But there is one major difference between them: formulation P_3 is made in such a way that all of the corner-feasible points are integer. It is fairly easy to see that if an LP relaxation is performed with this feasible region, independently of the objective function, the optimal solution will be an integer solution and consequently it will also be a solution to the MIP problem. Formulation P_3 is said to be *ideal* [40].

In most cases, since we are dealing with a large number of variables, there is no simple way to obtain this ideal formulation. So, instead of trying to find the ideal formulation, we may only want to know if, given two formulations P_1 and P_2, we can say that one is better than the other. Under certain conditions, we can [40]:

Definition 11. *Given a set $X \subseteq R^n$, and two formulations P_1 and P_2 for X, we say that P_1 is a better formulation than P_2 if $P_1 \subset P_2$.*

We will be particularly careful with the big M constants defined during the modeling techniques. The idea is to make them as small as possible to eliminate fractional solutions of the LP relaxation.

The MATLAB® function for MIP problems, as well as other software, runs Heuristic methods inside each node to quickly obtain good quality solutions. Optimality is not guaranteed for the search of these points, but they can help find better incumbent solutions (primal bounds), which in turn can increase the rating at which nodes are fathomed by bound. One of the best known methods is to define a neighborhood of the current incumbent solution and search for feasible points inside this neighborhood. It is know as local branching [12]. Another more recent procedure called Relaxation Induced Neighborhood Search (RINS) searches both inside a neighborhood of the incumbent and around the dual bound of the LP relaxation, with good results [10]. The function `intlinprog` offers three different possibilities for heuristic procedures. The first combines ideas from both methods presented above. The second uses exclusively RINS, and the third involves rounding the LP solution in a way that attempts to maintain feasibility. There is also the option of not using heuristic procedures at all, and only taking feasible points encountered during the branch and bound.

3.5 MULTI-OBJECTIVE OPTIMIZATION

Until this moment, and according to the formulation given for solving linear problems (at the beginning of Section 3.4), we considered that each optimization is done with a single objective function. This means that a sole criterion is either minimized or maximized. But there are many practical problems that involve decisions that must be taken in the presence of multiple objectives, sometimes even conflicting ones. For example, the decision of buying a computer maximizing performance, while minimizing cost, is a good example of a multi-objective problem. There are many techniques to solve these types of problem, *goal programming* being one of the most commonly applied. This alternative focuses on achieving certain target levels instead of maximizing or minimizing them. It models a usual assumption that the importance of any criterion diminishes once a target level has been achieved [29].

In this work, we are focusing on two other methods: *preemptive optimization* and *weighted sum of objectives*. Before explaining these methods, the concepts of *efficient point* and *efficient frontier* shall be defined:

Definition 12. *A feasible solution to a multi-objective optimization model is an efficient point if no other feasible solution scores at least as well in all objective functions and strictly better in one.*

Efficient points are also referred as *pareto optimal*.

Definition 13. *The efficient frontier of a multi-objective optimization model is the collection of the efficient points for the model.*

3.5.1 PREEMPTIVE OPTIMIZATION

Although an optimization may have multiple targets, normally they are not all equally important. Preemptive (or lexicographic) optimization explores this fact by taking them in order of priority. Optimization is performed considering one objective at a time. The most important one is optimized first; then the second most important objective is optimized subject to the constraint that the first one achieves its optimal value; and so on.

Mathematically this can be expressed in the following way: consider that a set of n optimization functions is in order of importance, so that f_1 is the most important and f_n the least important. Preemptive optimization consists in solving a sequence of single-objective optimizations of the form:

$$\text{minimize} \quad f_k(x) \tag{3.20}$$

$$\text{subject to} \quad f_j(x) \leq y_j^*, \quad j = 1, \ldots, k-1 \tag{3.20a}$$

$$x \in X, \tag{3.20b}$$

where $y_j^* := \min\{f_j(x) : x \in X\}$, i.e., y_j^* is the optimal value of the above problem, for $k = j$. This means that for each consecutive problem of the above form, an extra constraint is added.

One of the advantages of preemptive optimization is that the final solution is an efficient point of the full multi-objective model, i.e., it cannot be improved in one of the objectives without worsening another [29].

One major limitation (in some situations) is that it places a great emphasis on the first optimization. All the optimal solutions obtained in the subsequent single-objective problems are alternative optima of the first objective function. If these alternative optima are rare, there can even be cases where the total multi-objective optimization becomes essentially one of optimizing a priority objective, while neglecting all the others [29].

3.5.2 WEIGHTED SUM OF OBJECTIVES

An alternative to preemptive optimization is to combine the multiple objectives in a weighted sum. This procedure offers a more flexible handling of the different objectives than preemptive optimization. In the minimization of the composite function, weights of maximize objectives should be negative, and those of minimize should be positive.

Considering again a set of n optimization functions (now without strict priorities) $f_1, \dots f_n$, weighted sum optimization consists of a single-objective optimization of the form:

$$\text{minimize} \quad \sum_{k=1}^{n} w_k f_k(x) \tag{3.21}$$

$$\text{subject to} \quad x \in X, \tag{3.21a}$$

where the set $w_1, \dots, w_n$ represents the weights of the different objectives. It is also important to verify that the optimal solution to this problem is still an efficient solution. This comes from the evidence that any solution that could improve in one objective without degrading the others would also score better in the weighted objective. So only an efficient point can be optimal to the weighted sum.

CHAPTER 4

Targets and Constraints

In the previous chapter, it was pointed out that one of the main parts of an optimization problem is to define clear objectives (targets), as well as the conditions that the available choices must meet (constraints) in order to supply appropriate (feasible) solutions for the problem. This chapter is devoted to the study of possible targets and the constraints resulting therefrom.

After carefully analyzing the system presented in Section 2.2, some possible optimization targets can be identified. In the author's opinion, the most relevant are

- weight,
- three-phase power unbalance,
- further allocation possibilities,
- cost of implementation, and
- low priority shedding.

After discussion with SILVER ATENA, the first two were chosen as the most important and were defined as goals for this work. The third target is a good extra feature to have, and therefore it will also be discussed in detail.

Let us explore a bit more these three optimization targets. The first one refers to a specific criterion: *weight*. There are several advantages in optimizing (minimizing) the weight of the systems inside an aircraft. A lighter system is usually easier to transport, leading to lower transportation costs and consequently better performance. Weight can be measured in kg, for example, and therefore it is simple to state if an allocation scheme has a better or worse value of weight, when compared to another. Nevertheless, it is still necessary to define which parts of the system will contribute to the weight value. This will be analyzed in Section 4.2.

The second target, *three-phase power unbalance*, is a bit trickier. A system is considered balanced if the power necessary to supply the loads allocated in each phase is the same for the three phases. As it was already stated, balanced three-phase systems have better performance and also extend equipment's lifetime. Moreover, balanced systems are, in certain specific situations, a better solution if further three-phase loads must be connected (without changing the previous allocation). But when there is no possibility of having a perfect balanced system, how can we compare two solutions to state if one is better than the other? The answer to this and further unbalance concepts will be given in Section 4.1.

The *further allocation possibilities* target is related to possible late changes. In some situations, there can be a need to add loads to a previously optimized allocation scheme. This target is then associated with having the largest number of possibilities for further load allocations. For that, two main criteria must be taken into account: (1) the power consumptions and (2) the types of channels. If a load must be connected to one of the electrical phases, it is better for future allocations of three-phase loads that this load is allocated in the electrical phase with lower power consumption (this is not always true, as it will be discussed later). Also, if a load has a current rating, say, of 10 A, it is better (for further load allocations) to connect it to a channel supplying 7.5/10 A, than to a channel supplying 5/7.5/10 A. Allocating to the channel with fewer possibilities would enable the future connection of a load with either 5 A or 7.5 A or 10 A (while allocating to the first would only allow a further connection of a 7.5 A or 10 A load).

It has been specified already that the optimization of AC loads and DC loads can be done separately, according to the scope of decisions considered. However, the first target only makes sense for AC optimization, since DC has no electrical phases. The remaining two targets can apply to both AC and DC. The analysis of the target functions, as well as the constraints, is not going to be done separately for AC and DC, since they are basically the same. From now on, just keep in mind that the *three-phase power balance* only applies to AC optimization, along with all the constraints related to three-phase loads.

Before going into detail on these three optimization targets, we can briefly analyze the discarded targets.

Optimizing the *cost of implementation* would be to minimize the total cost associated with the allocation scheme. Several aspects can contribute to this: cost associated with the weight of the system (implementation, usage, etc.), cost associated with documentation changes due to modifications in system definition, and many others. Speaking in terms of engineering, this would clearly be the most relevant target. But for this purpose, it would be necessary to have access to some kind of cost estimations. These estimations are not available presently, and their development is outside the scope of the present thesis.

The *low priority shedding* target refers to the idea of distributing the sheddable devices in a way that a minimum number of high priority loads are disconnected when an overload in an PM feeder occurs. The benefits of this target were not conclusive, and therefore it is not going to be considered in this thesis.

At this point, it is worth remembering the formulation of an MIP problem (Equations (3.1)). The optimization involves a single linear objective function and it returns a single value. So every target must be formulated in a way that enables the calculation of a single result. Consider that we are trying to simultaneously minimize two parameters with a function returning two values instead of one, and that two feasible solutions are found. If both values are smaller for one of them, it is easy to classify it as a better solution. However, if one of the values is smaller and the other is larger, how can we assert that one solution is better than the other?

In this work we will first focus our attention in single-objective optimization. Afterwards, in order to optimize according to several targets, preemptive optimization is going to be the technique used most frequently. Some references to weighted-sum objectives are also made. Nevertheless, both of them also return a single value. The explanation of both these methods was given in Section 3.5.

4.1 THREE-PHASE POWER UNBALANCE

As mentioned previously, unbalance can have an impact on different components of the electrical three-phase network. In the distribution and transmission, the unbalance between currents will lead to additional power losses [5]. The evaluation of line loss under unbalanced systems and the propagation of the unbalance has been studied extensively [4], [5], [17], [19]. Regarding the electrical loads, it is important to refer to the AC induction motors since these are the most common motors used in aircraft applications [24]. Therefore, the decrease in performance, the heating and the shortened life of motors operating with unbalanced voltage/current is also an important subject. Several studies have been done regarding the response and performance of induction motors in an unbalanced system [2], [25], [39].

The major sources of unbalance are: transformers winding impedances, asymmetrical line impedance, but predominantly unbalanced distribution of single-phase loads. These single-phase loads consume different amounts of power, leading to unbalance currents in the system that in turn produce unbalanced voltage drops on the three phases of the supply system.

Currently, more flexible electrical grids are being studied in order to reduce the unbalance on the feeders and further improve the weight of the system. The weight can be reduced not only by decreasing the thickness and length of cables that directly supply the loads, but also by lowering the unbalance between the electrical phases. As discussed in Section 2.1.2, when the system is perfectly balanced no current flows in the return cable (neutral), which can allow cable thickness reduction. The concept of these new grids involves switching modules that enable loads to be shifted to a different electrical phase, during the operation of the aircraft [36]. In the present thesis, as explained already in the system's description (Section 2.2), loads are allocated to a specific channel (with predefined electrical phase) and cannot be switched to other channels or other electrical phases during the operation of the aircraft.

Ideally, the distribution of single-phase loads would be done in a way that would lead to equal loads in each phase, completely eliminating the origin of unbalance. However, most of the times this is not possible. For this reason, measurements of the degree of unbalance have to be defined so that different load distributions can be evaluated and compared.

4.1.1 DEFINITIONS

The degree of unbalance is usually defined using the symmetrical components method. This method simplifies the analysis of unbalanced three-phase power systems. The basic idea is that an asymmetric system can be broken down into [14]:

- *direct or positive-sequence components* consisting of three phasors equal in magnitude, displaced from each other by 120° in phase, and having the same phase sequence as the original phasors,

- *inverse or negative-sequence components* consisting of three phasors equal in magnitude, displaced from each other by 120° in phase, and having the same phase sequence opposite to that of the original phasors, and

- *homopolar or zero-sequence components* consisting of three phasors equal in magnitude and with zero phase displacement from each other.

We will not explain this method in further detail; more information can be found in [15].

VUP AND VUF

The most common methods to evaluate the degree of unbalance are the Voltage-Unbalance Percentage (VUP) and the Voltage-Unbalance Factor (VUF).

The first is based on the NEMA (National Electric Manufacturers Association) standard, and is given by [26]:

$$\mathrm{VUP} = \frac{\text{maximum deviation from average voltage}}{\text{average voltage}} \tag{4.1}$$

where voltages are measured from phase to neutral, and the average voltage is defined to be the average of the magnitudes of the three-phase to neutral voltages.

The second is based on the IEC (International Electrotechnical Commission) definition and uses the symmetrical components method [35]:

$$\mathrm{VUF} = \frac{\|\text{negative-sequence voltage}\|}{\|\text{positive-sequence voltage}\|} \tag{4.2}$$

This is a more meaningful definition; however, it is difficult to measure and calculate these sequence components without special instrumentation.

Both these methods do not take into account the phase angle, which contributes to the voltage unbalance. Hence, both definitions lead to errors in predicting, for example, the performance of induction motors [2]. Another more precise approach is an extension of the IEC definition and is defined as Complex VUF (CVUF) [37]. We will not go into further detail on this subject for reasons that will soon become apparent.

For the purpose of this work, it is more convenient to talk about current unbalance, since it is assumed that the generation and distribution of voltage is balanced. The unbalance will come from distinct current demands of the loads. Both the expressions given above can be used for this calculation by substituting voltage by current.

UNBALANCE DEFINITION FOR OPTIMIZATION

The relevant power informations, available for each load of the problem, are the

- Apparent Power, and
- Power Factor.

The Apparent Power of a load is the module of its Complex Power S:

$$|S| = \sqrt{P^2 + Q^2} \tag{4.3}$$

where P is the real power and Q is the reactive power (for a more detailed explanation of power in an AC circuit please refer to Appendix A).

The Power Factor is the cosine of the phase of the Complex Power:

$$\text{pf} = \cos\left(\tan\left(\frac{Q}{P}\right)\right) = \cos\theta \tag{4.4}$$

As stated in the previous section, we are mainly interested in the current unbalance. Since the voltage is a fixed parameter for the AC power system, current can be calculated by:

$$\mathbf{I}_{\text{rms}} = \left(\frac{S}{\mathbf{V}_{\text{rms}}}\right)^* \tag{4.5}$$

where "*" is the complex conjugate.

Alternatively, magnitude and phase can be calculated separately. The magnitude is calculated using the apparent power:

$$I_{\text{rms}} = \frac{|S|}{V_{\text{rms}}} \tag{4.6}$$

The phase is calculated by:

$$\theta_i = \theta_v - \theta \tag{4.7}$$

where θ_i and θ_v are the phases of current and voltage, respectively.

Now a problem arises: since only the apparent power and pf are known, how can we calculate the phase of the complex power θ? The answer is simple: we cannot. Notice that the pf, as defined in Equation (4.4), has the same value for two angles, since $cos(\theta) = cos(-\theta)$. To choose between one of these two values, additional information is needed.

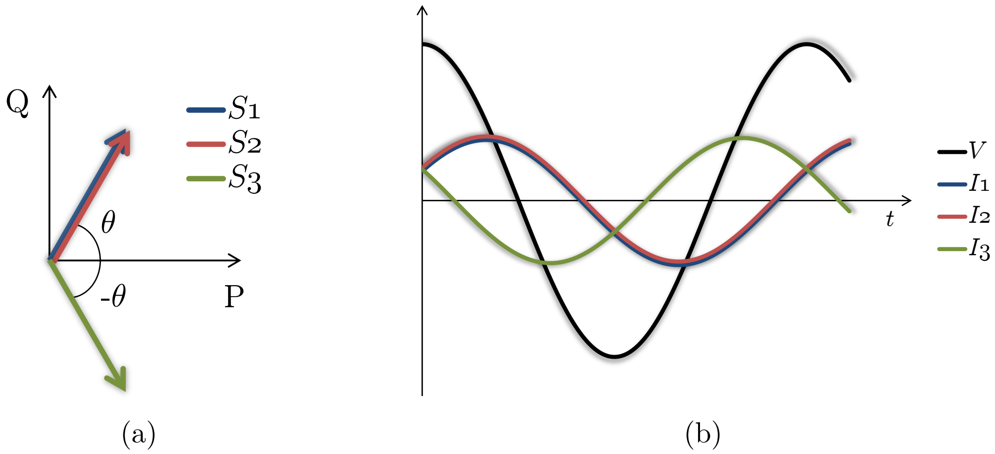


Figure 4.1: Example of power consumption from three AC loads. (a) shows the complex power vectors for the three different loads, and (b) depicts the variation of the current supplied to each load over time, considering the same voltage.

Figure 4.1 shows an example where the same apparent power and power factor can lead to different current consumptions and consequently to an unbalanced system. Figure 4.1a shows three complex power vectors (S_1, S_2 and S_3, corresponding to three different loads), with the same module (apparent power) and the same power factor ($\cos(\theta) = \cos(-\theta)$). However, the phase of S_1 and S_2 is different from the phase of S_3. These power vectors lead to the currents depicted in Figure 4.1b. None of the currents is in phase with the voltage, due to the fact that the loads are not purely resistive ($\theta \neq 0$). Yet, if Loads 1 and 2 are connected to different electrical phases, these phases will be balanced. The same does not happen with Load 3, due to the different power phase angle.

Thus, unbalance cannot be calculated using only the information given. To further address this problem, some simplifications are made. First, it will be considered that all the loads have pf = 1. This way, and since we are assuming balanced voltages, all currents will be in phase with the voltage on each electrical phase, and 120° apart between each phase. Note that the power limits are always satisfied, despite this simplification: the assumption of a zero angle always leads to the largest possible value of the total complex power (vector summation of all loads). Hence, we ensure that the actual apparent power consumption is never greater than the one considered. In the worst case scenario, we are ignoring some possible allocation schemes, but at least the safety of the system is not compromised.

Analysis of the test data used in this thesis shows that the considered approximation is not completely unreasonable, since most of the power is concentrated in power factors close to 1. However, there is also a considerable amount of power consumptions with power factors down to 0.85. This corresponds to a complex power's angle of approximately ±30°, which is not negligible. It could be the case that most of this power would come from three-phase loads, leading to an equivalent affection of all electrical phases. Unfortunately this cannot be guaranteed, meaning that the results from balancing calculations must be regarded with special care.

The calculation of unbalance using Equation (4.2) involves exact information about the current phasors. Considering that the values given (operational power) are averaged values and that we are making further simplifications, it does not make much sense to use such an accurate expression. Therefore, Equation (4.1) proves to be a more pragmatic approach, as phasors are not taken into account.

Since we are considering balanced voltages (i.e. they have the same magnitude and are 120° apart between each phase), the calculation of the current unbalance is identical to the calculation of the apparent power unbalance, to a difference of a constant factor (see Equation (4.6)). This suggests an easier approach to the unbalance calculation, using the available data, through a modification of Equation (4.1):

$$\mathrm{Unb}_1 = \frac{\text{maximum deviation from average apparent power}}{\text{average apparent power}} \tag{4.8}$$

where the average apparent power is the mean value over the apparent power consumptions on the three phases.

Two additional methods for unbalance calculations were considered, based on previous experience with customization. The first one takes into account the differences between the powers on each phase:

$$\text{Unb}_2 = max(|P_A - P_B|, |P_B - P_C|, |P_C - P_A|) \tag{4.9}$$

where P_A, P_B and P_C are the apparent power consumptions on phases A, B and C, respectively. This means that the unbalance is taken as the maximum difference of power between two phases. Note the choice of the letter P, instead of $|S|$. This comes from the assumption that the power factor is equal to 1 (P=$|S|$, for each electrical phase).

The other method has some similarities with Equation (4.8), and is defined as:

$$\text{Unb}_3 = \frac{max(|P_A - P_B|, |P_B - P_C|, |P_C - P_A|)}{P_A + P_B + P_C} \tag{4.10}$$

This calculation has an advantage (also present in (4.8)) when compared to (4.9). Imagine that we have one optional load to allocate and two feeders with different power consumptions from the previous allocated standard loads, as depicted in Figure 4.2.

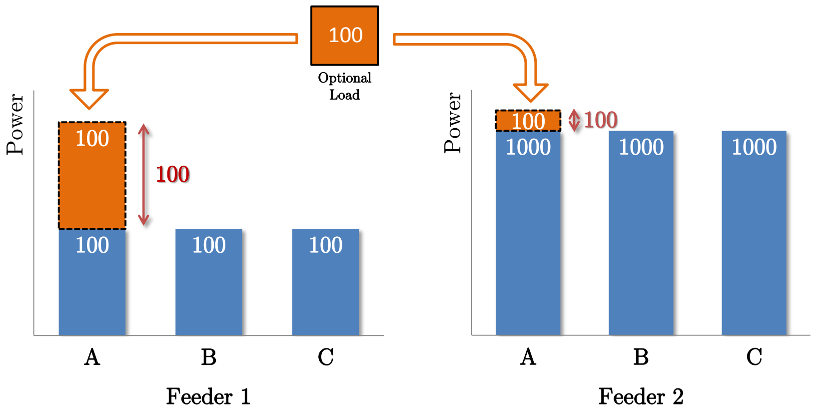


Figure 4.2: Allocation of a load to feeders with different amounts of power consumption.

Both solutions have the same unbalance (100 VA) when Equation (4.9) is used. Nonetheless, it should be evident that the allocation of this optional load to the feeder with considerably larger power consumption will have a lower impact on the operation of the corresponding system. Using (4.8) or (4.10), the values for the unbalance are given by:

Feeder 1

$$\text{Unb}_1 = \frac{200 - 133.\underline{3}}{133.\underline{3}} = 0.5$$

$$\text{Unb}_3 = \frac{100}{400} = 0.25$$

Feeder 2

$$\text{Unb}_1 = \frac{1100 - 1033.\underline{3}}{1033.\underline{3}} \approx 0.065$$

$$\text{Unb}_3 = \frac{100}{3100} = 0.032$$

The results show that the choice of allocating to the second feeder would lead to an unbalance value approximately eight times lower than the allocation to the first feeder, using these two definitions. Therefore, these calculation methods would select, correctly, the allocation to the second feeder as a better solution, while (4.9) would classify these solutions as identical.

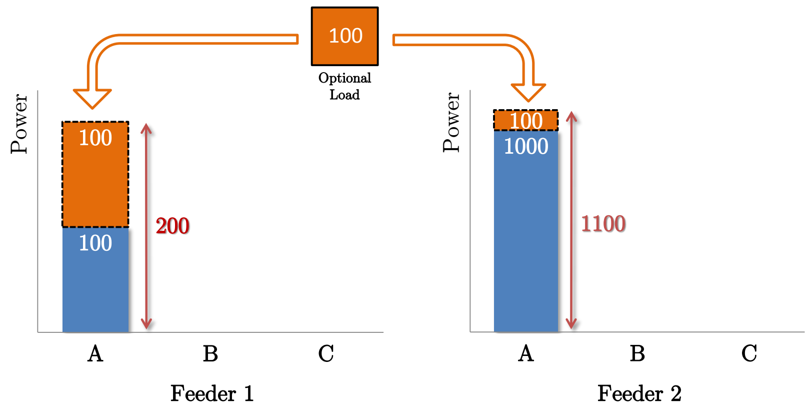


Figure 4.3: Allocation of a load to feeders containing power only on one electrical phase.

Still, there is a specific situation where (4.9) will attain a better performance than (4.8) and (4.10). An example is given in Figure 4.3. Here the optional load, due to some contraints (for example, current rating), can only be allocated to electrical phase A, this being the only phase consuming power. Comparing the two possibilities given, the allocation to the first feeder would be, in principle, a better option, since we do not want to worsen what is already the largest unbalance. The problem with definitions (4.8) and (4.10) is that when the power is only consumed by one phase the outcome is constant, that is, it does not depend on the amount of power:

$$\text{Unb}_1 = \frac{\text{maximum deviation from average power}}{\text{average power}} = \frac{P_A - \frac{P_A}{3}}{\frac{P_A}{3}} = 2 \tag{4.11}$$

$$\text{Unb}_3 = \frac{max(|P_A - P_B|, |P_B - P_C|, |P_C - P_A|)}{P_A + P_B + P_C} = \frac{P_A}{P_A} = 1 \tag{4.12}$$

Nevertheless, this type of situation is not common in the present system, so this disadvantage would have no real impact.

After studying these three possibilities for unbalance calculation, Equation (4.9) was chosen for the optimization, despite the fact that the other two lead, in general, to better results. The effort needed for the implementation of these target functions would be considerably greater and most of the time, these would yield the same results as the first one, since, for many systems, most of the optimizations can be performed for each feeder separately. Therefore, the total power ($P_A + P_B + P_C$) is a constant, meaning that the target functions (4.9) and (4.10) lead to the same optimization result (in terms of allocation). With a fixed amount of power, the unbalance Equation (4.8) also does not prove to be clearly better than Equation (4.9), since its main advantage (previously explained) vanishes.

4.1.2 TARGET FUNCTIONS

So now we have defined a clear method to calculate the unbalance:

$$\text{Unb} = max(|P_A - P_B|, |P_B - P_C|, |P_C - P_A|) \tag{4.13}$$

where P_A, P_B and P_C are the apparent power consumptions on phase A, B and C, respectively.

But, looking at the problem, we have operational and maximum power consumptions, intermittent and permanent operation, sheddable and non-sheddable loads, different flight phases, different feeders, different cable segments... How can we deal with all this?

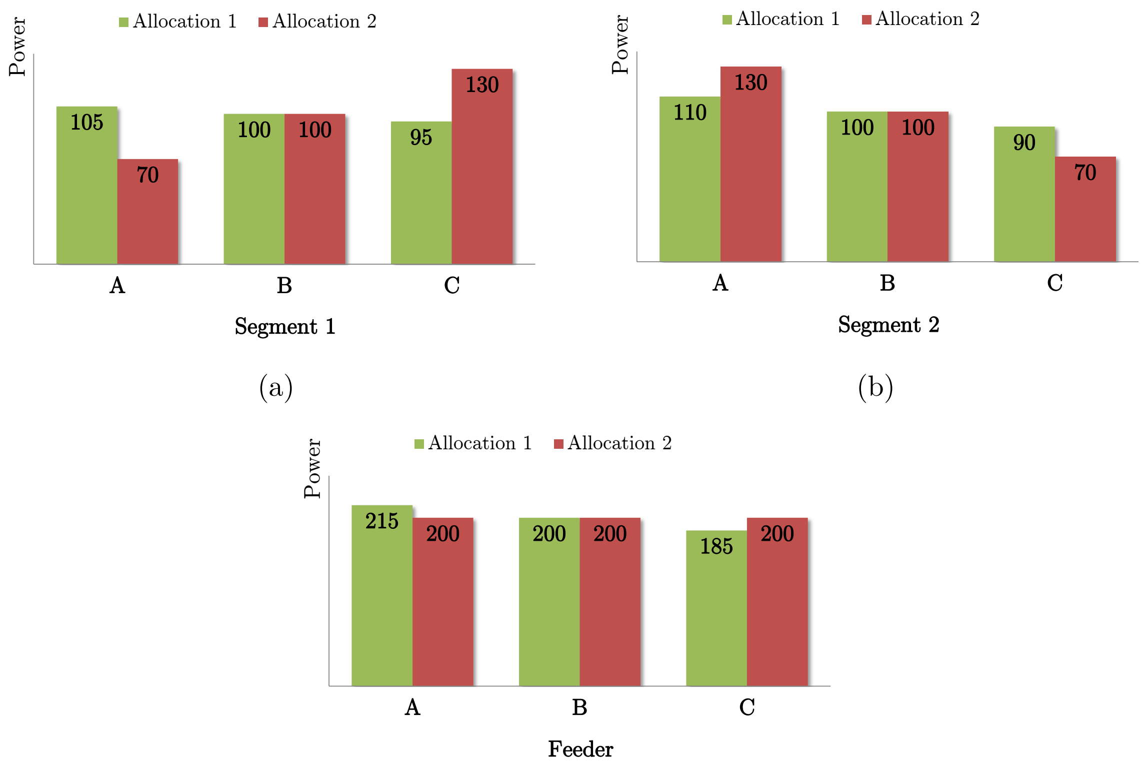
Beginning with the types of power, it was stated in Section 2.2.3 that the loads' power consumptions in most flights around the world are represented by the operational power consumption, being the maximum power the consumption under the worst conditions. Hence, even though applicable limits shall be verified for both types of power consumption, the optimization's target function will only consider the operational power.

Regarding the operation mode (see 2.2.5), there is normally no way of predicting the intermittent consumptions, at least for some of the loads. These consumptions can occur at different instants in time for each of the loads. Therefore, only the power coming from permanent operation shall be optimized. Once again, do not forget that the intermittent consumptions still have to be considered for the applicable limits' calculation, as they take part in the constraints.

Concerning the types of loads (see 2.2.6), in a normal situation, all loads will be operating normally. Sheddable loads are only shut when overloads occur. Thus, all the loads shall be considered for the optimization targets.

Moving on, there are still a few things to discuss: flight phases, feeders and cable segments. Let us start by neglecting the flight phases and considering only the different feeders and cable segments. In the beginning of this section, it was pointed out that the line power loss of unbalanced systems, as well as the propagation of unbalance throughout the circuits, are consequences of the loads' distinct power consumptions. This indicates that the balancing should be performed as early as possible, that is to say, on each box (SPDB level). This way the cable segments directly connected to the boxes will be as balanced as they can.

But looking at the big picture, optimizing a step higher can enable a better overall balancing, by compensating with the different cable segments of the feeder. This in turn can diminish the stress to the generators and essential motors needed for aircraft operation. Figure 4.4 shows an example where the second allocation, although worse for each of the cable segments, leads to a better balancing on the feeder level.



(a)

(b)

(c)

Figure 4.4: Benefit of optimizing on the feeder level, instead of box level.

The same argument could be used to support the three-phase balancing at higher levels, closer to the power generators. However, this is out of the scope of this thesis, since it would involve systems outside the cabin and cargo distribution. Nevertheless, if data of the remaining system is available, the developed optimization tool can also be adapted to this situation. Taking this into account, the balancing optimization will be carried out at the feeder level. Again, we are not neglecting the cable segments, their limits must be considered in the constraints.

There are several situations where feeders can be optimized independently. But, for the cases where this is not possible, and since single-target optimization, as the name suggests, can only maximize or minimize a specific linear function, it is important to think of possible ways of evaluating an allocation when multiple feeders are involved.

The same principle applies to the different flight phases. Remember that, depending on the flight phase, loads have different power consumptions, but the allocation is unique for all flight phases. What will be the target of the optimization? The unbalance on a specific flight phase? The maximum unbalance over all flight phases?

The duration of each flight phase depends on the type of flight. For example, an intercontinental flight can have a long cruise phase when compared to the ground operation, while for a domestic flight, these phases can have almost the same time. It could be, in theory, possible to ask the airline company what would be the main purpose of the aircraft in terms of the flight's type, but normally airline companies use the same aircraft for different types of operations.

Three different methods were considered in order to obtain a single unbalance value for the optimizations of all the feeders/flight phases. These are: (1) mean value, (2) maximum value and (3) mean squared value.

We have a set of values, what is the first idea that comes to mind (at least the author's mind) to get a single value? Average! So the first method is probably the most intuitive. It has the clear advantage of taking into account the influence of all the values. On the other hand, it has the disadvantage of being unable to give information on the largest value, which can lead to critical situations.

The second method comes from the simple idea of minimizing the worst case. Still, the disadvantage of this method is exactly the advantage of the mean value, i.e. only looking at the maximum value can rule out solutions with the same maximum value, but lower overall unbalance.

The last one comes as a sort of improvement to the mean value. The idea is to calculate the mean of the squared values. The square value penalizes large values of unbalance, compensating the disadvantage of the mean value approach. This is inspired on the well known mean squared error (MSE).

Table 4.1 shows a possible outcome of three different allocations on a system with four feeders/flight phases and the resulting values using the three definitions.

	Allocation			
	1	2	3	4
Feeder/ Flight Phase	Unb	Unb	Unb	Unb
1	50	50	150	80
2	50	0	0	60
3	50	0	0	0
4	50	0	0	0
Max	50	50	150	80
Mean	50	12.5	37.5	35
Mean Squared	2500	625	5625	2500

Table 4.1: Comparison of the different unbalance calculations when multiple feeders or multiple flight phases are involved.

The first and second allocations expose the drawback of considering only the maximum value. The second allocation is clearly better, but for a target function considering the maximum, both the allocations would have the exact same value.

Comparing now the first and third allocations, we will consider the first a better allocation scheme, since it reduces a considerably higher unbalance on one of the feeders/flight phases. This decision is based on an engineering judgment made jointly with SILVER ATENA. If a mean target function would be used, allocation three would be "incorrectly" selected as a better scheme.

Notice that the results of the mean squared for the first three allocations indicate that this target function would, on its own, choose the "correct" allocation in both comparisons.

The problem starts when the values are not so disparate. Comparing allocation schemes 1 and 4, both have the same mean squared value, but the first one has a lower maximum value, while the fourth has a lower average. When the values tend to get closer, the decisions become harder. It will be assumed that the first allocation is better than the fourth, therefore, a first optimization on the maximum value will be used, followed by a further optimization on the mean value, with max as a constraint. More information will be given later.

As already discussed, since airline companies can usually use the same aircraft for different types of operation, it is difficult to predict the importance of each flight phase during the operation of an aircraft. However, in order to provide the possibility to assign different degrees of importance to each flight phase, an optimization target with a weighted sum of the unbalances over the flight phases is also developed. As it will be shown later, this target can be closely linked to the mean unbalance optimization.

Regarding the flight phases, an extra optimization target will be considered. The idea is to consider the power on each electrical phase as the maximum among all the flight phases. The unbalance is calculated using these values and the final result is the maximum over all feeders. In terms of the performance of the system, this function can lead to strong unbalance situations even if the result of the optimization is nearly zero. The author decided to still consider this target, since it was strongly used for verifying the implementation of the mathematical model and it has some additional interest in terms of modeling.

Summarizing, the targets are:

1. Minimize the maximum unbalance among all feeders and all flight phases.
2. Minimize the mean unbalance among all feeders and all flight phases with the possibility of assigning weights to the flight phases.
3. Minimize the maximum unbalance among all feeders, considering the powers on each electrical phase as the maximum among all flight phases.

Table 4.2 shows an example of the results obtained with these three different targets. Notice the referred problem of the third target on the first feeder.

Feeder	Electrical Phase	Flight Phase 1	Flight Phase 2	Max
1	A	100	400	400
	B	400	150	400
	C	400	50	400

Unbalance	300	350	0
Max	350		
Mean	325		

Feeder	Electrical Phase	Flight Phase 1	Flight Phase 2	Max
2	A	100	200	200
	B	200	50	200
	C	200	250	250

Unbalance	100	200	50
Max	200		
Mean	150		

Optimization Target	Value
1	350
2	237.5
3	50

Table 4.2: Example of the results obtained for the unbalance, considering each of the optimization targets.

4.1.3 CONSTRAINTS

The constraints can be derived from the system analysis performed in section 2.2. If we consider that the cards in each box are already defined, i.e., no card selection is allowed, the constraints are:

1. Each optional load can only be allocated to its predefined box.

2. Each optional load can only be allocated to channels that supply the necessary current rating and that are not already taken by standard loads.
3. Each channel cannot supply more than one load.
4. Each load must be allocated to one channel.
5. All feeders and cable segments must comply with the applicable limits.

An extra constraint is added if the 3-phase loads have a specific connector, as described in Section 2.2.2:

6. The three phases of a 3-phase load must be connected to three consecutive channels, beginning in phase A.

If cards can be chosen for free card slots, or if card changing is allowed:

7. Only a single type of card must be selected for each card slot (leaving it empty is also an option).

4.2 WEIGHT

Weight reduction is a constant ambition of aircraft manufacturers and airline companies, since it is regarded as one of the ways to increase profit. Some research is directed at changing the commonly used 115 VAC to higher voltages, such as 230 VAC or 270 VDC [6], [8], [9]. This would enable thinner cables and consequently lower the weight of the electrical system. Other topics in weight-saving research involve changes to the architecture of the electrical distribution system, namely the relocation of the central power center or parts of it [32]. Power management techniques have also been studied in order to reduce current ratings of the different feeders composing the electrical network [31].

The present work focuses on the cabin and cargo distribution. There can be potential for weight savings, depending on the commercial loads' allocation choices. It is common sense that a cable with higher current rating entails higher weight, hence the lower the current necessary on a certain cable, the lower the necessary current limit and consequently the lower the weight. If power consumption is distributed in an optimized way, cable ratings may be reduced, leading to weight savings.

Moreover, as it was discussed before, it is sometimes possible to choose the types of LRM cards. These cards may have different weights, and thus their choosing can also affect the weight of the system.

It is considered that there is only a set of possible values for the cable current rating, and consequently for its power limit (voltage is fixed). The length of the wires is also fixed. Thus, the weight of the cable as a function of the total load's current can be represented by a left-continuous staircase function, as shown in Figure 4.5. It is left-continuous due to the fact that the maximum current for a given cable can still be supplied by it.

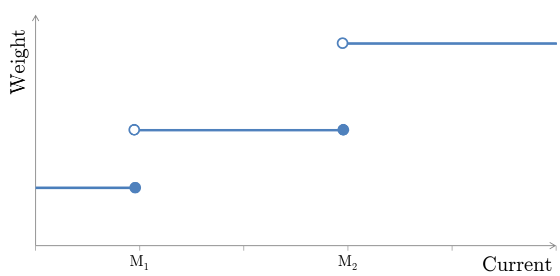


Figure 4.5: Sketch of cable weight against current supplied. M_1 and M_2 represent the maximum current that the first two cables can supply.

Although both cable and card choices contribute to the weight, they imply changes on different levels of the system. Card choices can be regarded as a frequent procedure, and are normally easy to perform. Cable choices involve changes beyond the SPDB boxes, and are a less common and more complex procedure.

When phase balancing optimization is performed, it is possible to allow card choices, as we have already discussed. Since this optimization does not take into account the weight of the system, there can be situations where, for example, one extra card is installed without improving performance, when compared to other solutions. Further weight optimization, considering only the cards, would solve this problem. It makes sense then, to have the possibility of optimizing weight, considering only card choices and leaving the original cable setup.

4.2.1 TARGET FUNCTIONS

For the reasons given, two optimization targets will be considered:

1. Minimize the weight of the system considering cables and cards.
2. Minimize the weight of the system considering only cards.

4.2.2 CONSTRAINTS

The constraints presented in Section 4.1.3 for the three-phase power balance also apply to the weight optimization. Only an additional constraint must be considered, regarding the cable choices:

8. Only a single type of cable must be selected for each supplying cable (leaving it empty is not an option).

4.3 FURTHER ALLOCATION POSSIBILITIES

There can be situations where loads must be connected to a previously optimized allocation scheme, without changing the connection choices already made. The idea is to enable further load allocations if desired in the future. For this, one of the things to consider is the amount of different ratings each channel can supply. It could be possible to study the amount of loads that need each type of current rating, and perform a probabilistic analysis, in order to try to predict which current ratings are the most used. In this work, it is considered that there is no preferred current rating. Hence, a channel with more possibilities for current ratings (regardless of their value) provides more opportunities for future load allocations.

Another aspect that must be taken into account is three-phase loads' allocation, specifically cases where the different phases must be connected together. In these situations, in order to allow future allocations, all three channels of a group must be free (more details in Section 2.2.2) and contain the required current rating.

This optimization objective is regarded as secondary when compared to the previous two. Therefore, no optional selection of cards will be considered. It is assumed that the optimization was already performed with respect to another target, which led to the definition of the optional cards. Also, if the choosing of cards was allowed, it would raise problems with the decisions for this objective function, since cards with more channels would always have an advantage, when compared to cards with fewer channels.

4.3.1 TARGET FUNCTIONS

For deriving the target function, each channel will have a value attached corresponding to the different number of current ratings it can supply. To take into account the three-phase loads, to each group of three consecutive channels beginning in phase A we assign the number of current ratings that they all can supply.

1. Minimize the number of used ratings considering single-phase and three-phase loads.

4.3.2 CONSTRAINTS

The constraints from Section 4.1.3, excluding the one dealing with optional cards, also apply to this type of optimization.

4.4 FURTHER CONSIDERATIONS

4.4.1 RELATIONSHIPS BETWEEN TARGETS

THREE-PHASE BALANCE VS. WEIGHT

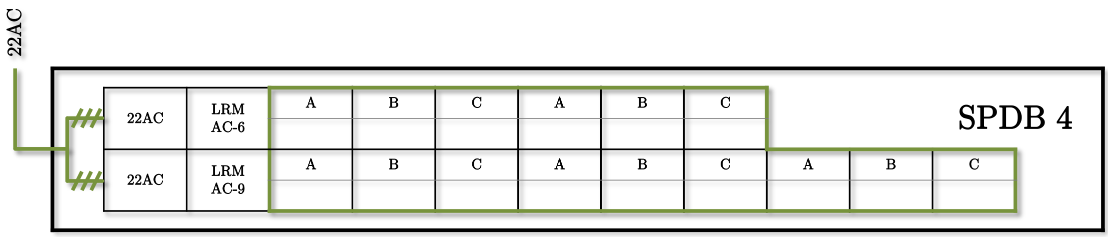


Figure 4.6: Single cable supplying a set of channels.

Single Cable. Let us look into the optimization targets in more detail. Consider the case of only one feeder supplying a set of channels (see Figure 4.6), and only one applicable limit. As discussed in Section 2.2.7, the applicable limits shall be met for all the electrical phases, that is, the power supplied to each phase must be below these specified limits. If not, a larger (and consequently heavier) cable must be used. Clearly, the best way to keep the power below the applicable limit would be to distribute it evenly over the three phases. Hence, a solution for the three-phase power balance optimization is in fact also a solution for the weight optimization. And the opposite? Can we assume that a solution for the weight optimization is also a solution for the three-phase balancing? The answer is no. Figure 4.7 shows results from two possible allocation schemes.

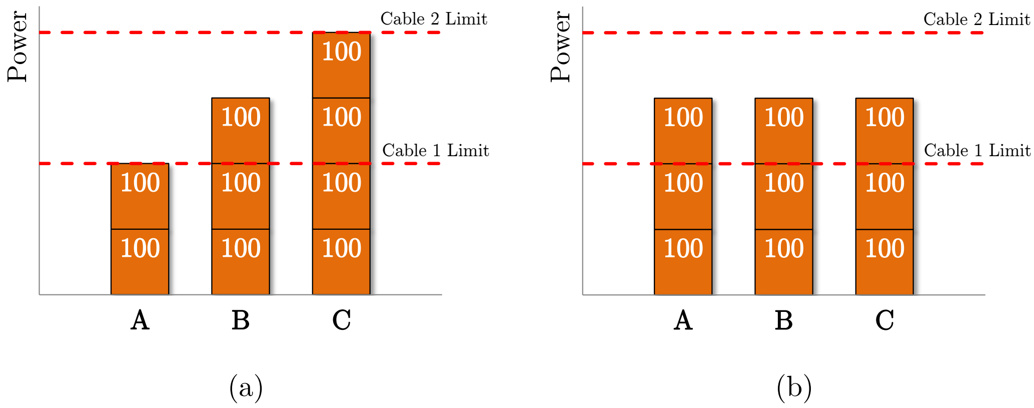
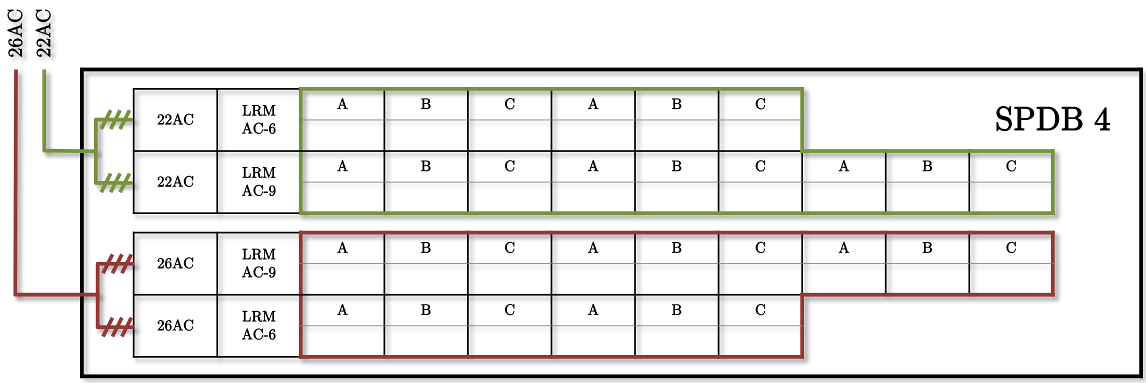
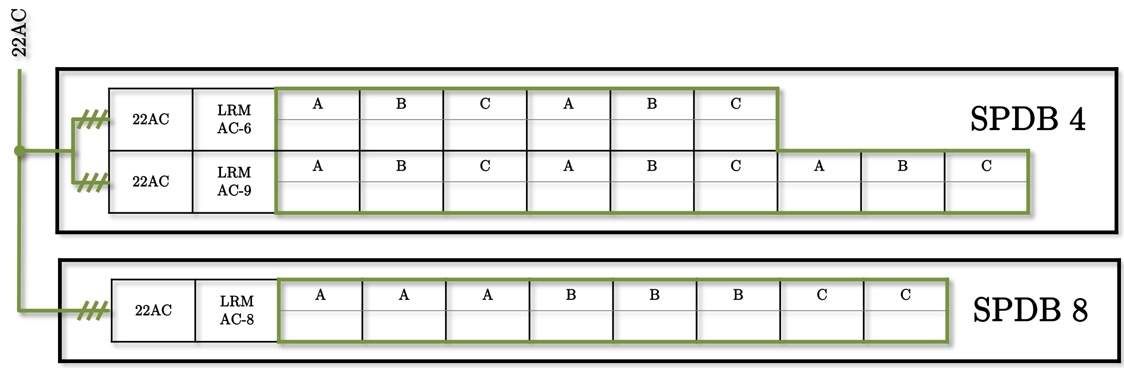


Figure 4.7: Two possible allocations with a single cable supplying a set of channels. (a) is only an optimal solution for weight optimization, while (b) is an optimal solution for both weight and phase balancing optimizations.

Remember that the result from the optimization problem is a single value, therefore, both results attain the same weight for the system, since both need the same cable type. However, the first is not an optimal solution for the three-phase balancing problem, since there is at least one allocation with a better value for the optimization target. The second can be proved to be an optimal solution, since the unbalance can never be smaller than zero. This leads to the conclusion that when we are optimizing the allocation for one cable, there is no advantage in optimizing the weight instead of the power balance (when all cards are fixed).



(a)



(b)

Figure 4.8: Two cables supplying a set of channels. (a) depicts two feeders connected to the available slots and (b) depicts one feeder supplying loads through different cable segments.

Multiple Cables. Can we extend this conclusion if more than one cable supplies the available slots (see Figure 4.8)? Figure 4.9 depicts an example where a single optional load must be allocated (this example is related to the system of Figure 4.8a, but the same is valid for Figure 4.8b). Even though the allocation of the optional load to the first cable would lead to a global decrease in the unbalance of the system (it is easy to see that this is the optimal solution), due to the proximity to the applicable limit, this allocation would require a larger cable type to supply the first card and a consequently worse weight value. Therefore, the optimization of the power balance will not necessarily lead to the optimization of the weight, as it was verified for the single cable case.

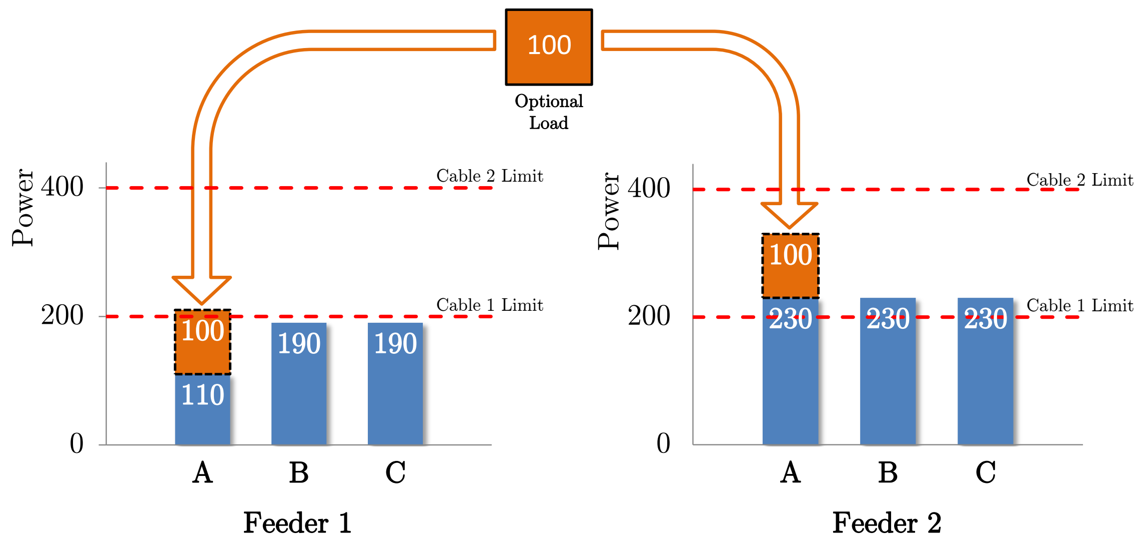


Figure 4.9: Difference between weight and phase balancing optimizations. The allocation to feeder 1 leads to the (single) optimal solution for phase balancing, but not for weight. Allocation to feeder 2 gives an optimal solution for weight, but not for phase balancing.

Optional Cards. If optional cards are considered, these two targets can have different results even in the case of a single cable. Consider the example of Figure 4.10.

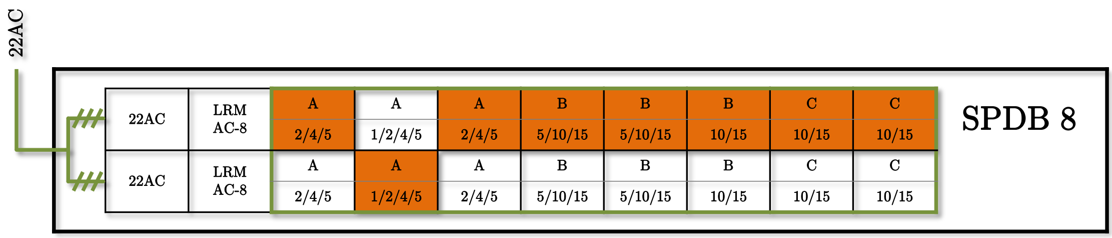


Figure 4.10: Single cable supplying a set of channels. With optional cards, weight optimization should be performed after phase balancing optimization.

Suppose that this is the balancing optimization result considering that both cards are optional. Clearly there is a better allocation scheme in terms of weight with the same power balancing. The load allocated to the second card can be allocated to the second channel of the first card. Since both cards are equal (these channels allow the same current ratings), and both are supplied by the same feeder, other constraints are not a problem. So further optimization on weight, with fixed unbalance, would lead to the best balancing with the minimum possible value of weight.

Optimizing first on weight, although ensuring the lightest system (only one card is used), would not mean that we have the best balancing with one card. A further balancing on optimization would be needed.

In conclusion, only in the case of one cable and no optional cards will both optimizations lead to the same allocation. Since this case is not a frequent situation, when optimizing weight, a further optimization on balancing should be performed, using preemptive optimization. When optimizing electrical phase balancing, a further optimization on weight should be used.

Single Cable. In the beginning of this chapter, when the optimization targets were chosen, the author stated: "Moreover, balanced systems are, in general, a better solution if further three-phase loads must be connected (without changing the previous allocation)". This is true if we consider a scenario like the one depicted in Figure 4.6. Consider the connection of a three-phase load to a previous allocation scheme illustrated in Figure 4.11.

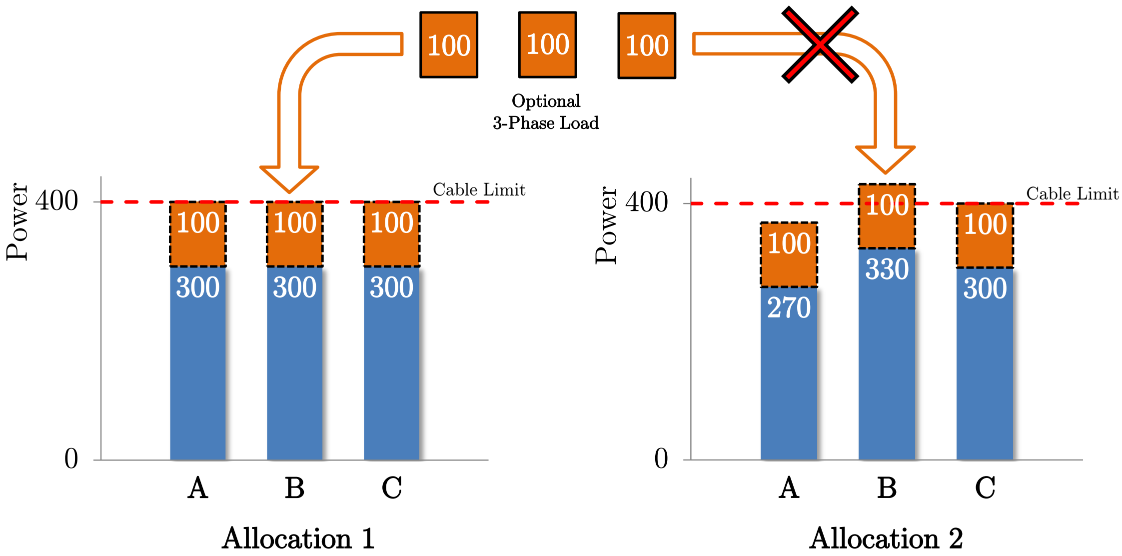


Figure 4.11: Two possible allocations with a single feeder. Balanced system proves to be a better solution for further allocation of three-phase loads.

Since the three-phase load consumes the same power over the three-phases, a perfectly balanced system would in fact be the best solution for a further three-phase load allocation. And single phase loads? Unfortunately, further single phase load connections would benefit from an unbalanced system. Figure 4.12 shows a situation where a single phase load can only be allocated if the system is not previously perfectly balanced.

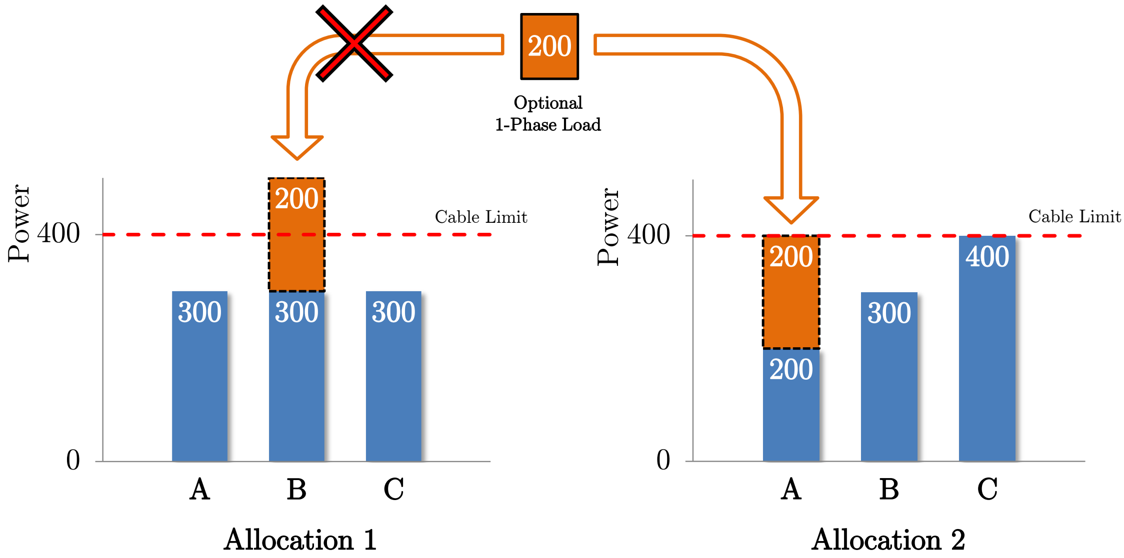
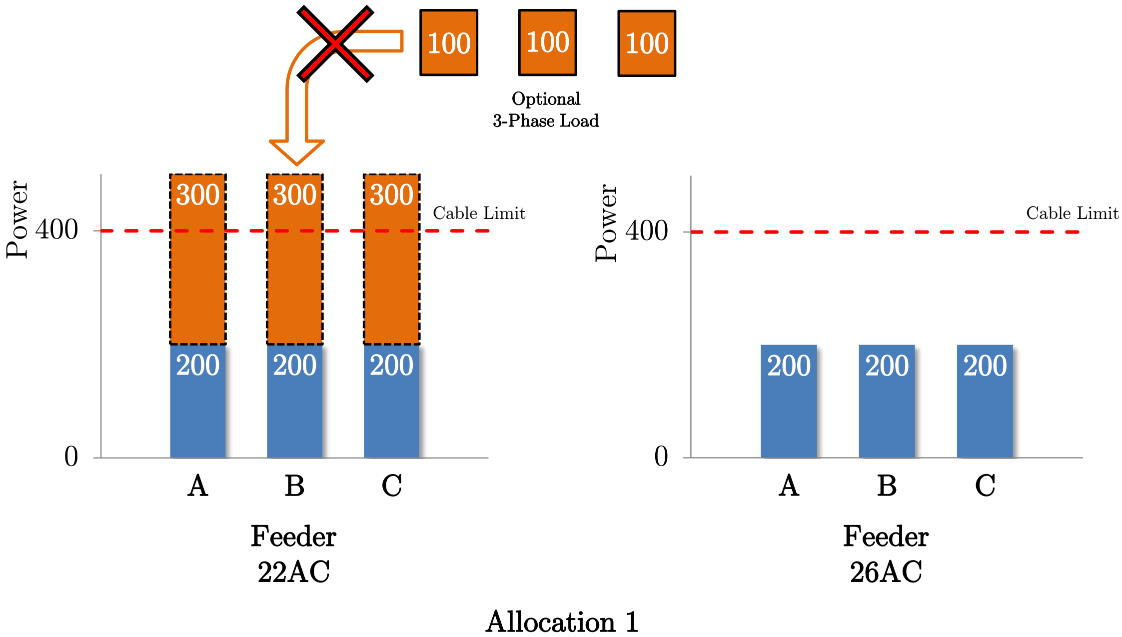
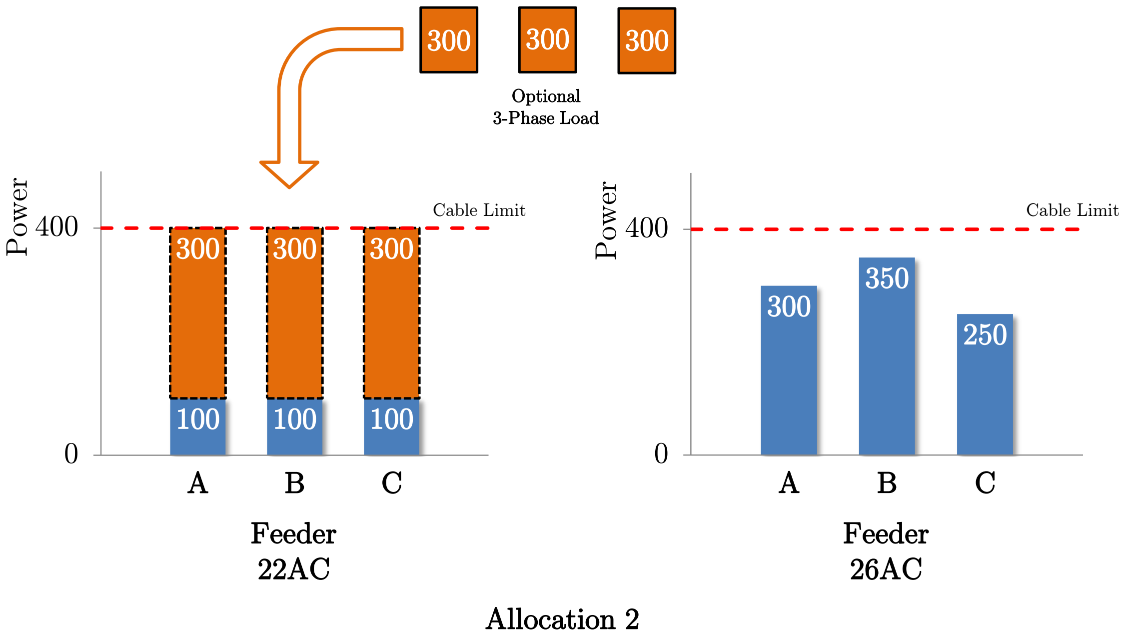


Figure 4.12: Two possible allocations with a single feeder. Unbalanced system proves to be a better solution for further allocation of single-phase loads.

Multiple Cables. If we extend this analysis to the situation with more than one cable (see Figure 4.8), even the further allocation of three-phase loads does not necessarily benefit from the best balancing solution. An example is given in Figure 4.13



(a)



(b)

Figure 4.13: Two possible allocations with two feeders. Balanced system no longer implies a better solution for further allocation of three-phase loads.

Basically, this shows that the phase balancing optimization is not necessarily correlated to the optimization of further load allocation possibilities.

CHAPTER 5

Mathematical Model

With the targets and constraints properly defined, we now focus our attention on developing mathematical models that can be used to reach optimal solutions. This is done in the case of AC optimization, since it is the most complex and the one that involves all the optimization targets. However, at the end of this chapter, a short paragraph explains the necessary changes applied to DC.

The model for the complete problem is presented below, but this does not mean that it was all developed at the same time. In a complex problem like the present one, it sometimes helps to start with smaller problems and then gradually add complexity, until the total model is obtained. When following this procedure, debugging also becomes easier, since each step can be tested separately. This is somewhat similar to the development of a complex computer program.

5.1 DEFINITIONS

The problem to be considered is how to allocate a set of optional loads L to a set of channels S. The number of optional loads is denoted as N_L and the number of channels as N_S. Observe that, based on the problem's restrictions, these two values must obey: $0 \leq N_L \leq N_S$. The channels belong to different boxes and cards, and have specific current ratings (see example in Figure 2.9). Moreover, some of them may already be occupied by standard loads. For these reasons, as it was discussed in Section 2.2, not every channel is available to each optional load. Every load has a specific box where it must be connected, and also a predefined current rating that must be supplied by the channel (see also Sections 4.1.3, 4.2.2 and 4.3.2 for the constraints summary). Only one load can be supplied by each channel, which means that the ones already occupied by standard loads are not available. Regarding this, we define the sets $S_l \subseteq S$ with the channels where load $l \in L$ can be allocated, and $L_s \subseteq L$ as the set of loads that can be allocated to channel $s \in S$.

The binary variables $x_{l,s}$ represent the allocation of load l to channel s. We set $x_{l,s}$ to 1, if load l is connected to slot s, and to 0 if otherwise. These are the variables representing the allocation decisions.

As it was presented in the system's definition, there are different flight phases and each load is characterized by four types of power per flight phase, based on operating mode (permanent and intermittent) and on the type of power (operational and maximum). Let us consider that the system has N_Γ flight phases and the powers are given by: $p_{l,\gamma}^{\mathrm{o,per}}$, $p_{l,\gamma}^{\mathrm{o,int}}$, $p_{l,\gamma}^{\mathrm{m,per}}$ and $p_{l,\gamma}^{\mathrm{m,per}}$, where l identifies the load and γ is the number of the flight phase. In the superscript, o and m represent the type of power (operational and maximum, respectively), while *per* and *int* stand for permanent and intermittent operation.

There is another important distinction between the loads, related to power management procedures. Specifically, loads can be either sheddable or non-sheddable (see Section 2.2.6). Let L_{sh} be the set of sheddable loads and $L_{\overline{\text{sh}}}$ the set of non-sheddable loads. Loads can only be sheddable or non-sheddable, so the union of these two sets has N_L elements (all loads) and their intersection is the empty set.

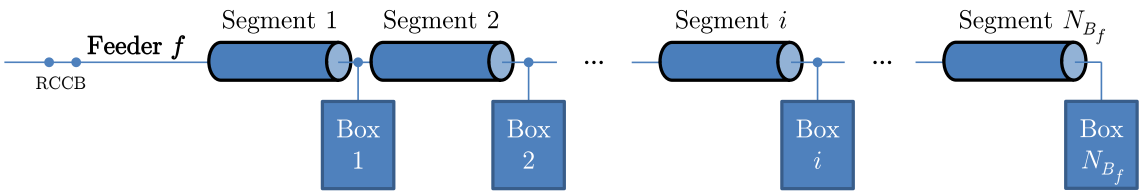


Figure 5.1: Segments and boxes of a feeder's network.

The channels are allocated to different boxes, connected to distinct feeders. As presented in Figure 5.1, for a given feeder, each box has a certain number of upstream (direction of power source) cable segments, depending on the position in the distribution (a more detailed explanation is given in the system's description in Chapter 2). These segments are all supplied by the same feeder, but it is considered that they can have different power limits. Nevertheless, it is notorious that, for instance, segment 1 must have a power limit equal or greater than segment 2, since all the power that goes through segment 2 also goes through segment 1. The boxes are going to be referred to by the number of the immediate upstream segment. Figure 5.1 illustrates this. Consider F as the set of feeders of the problem, and N_{B_f} as the number of boxes and, consequently, segments connected to feeder $f \in F$. Let then $S^{\phi}_{f,b} \subseteq S$ be the subset of channels supplied by feeder $f \in F$, box $b = 1, \ldots, N_{B_f}$, and electrical phase $\phi = 1, 2, 3$ (A, B, C, respectively). The index $b = 0$ is used to refer to the total set of slots supplied by the feeder. Note that:

$$\cup_{b=1}^{N_{B_f}} S^{\phi}_{f,b} = S^{\phi}_{f,0}$$

and

$$\cap_{b=1}^{N_{B_f}} S^{\phi}_{f,b} = \emptyset$$

As previously stated, the applicable limits depend on the type of feeder (with PM or without PM), but also on the relation between the power limit of the cable segments (this is related to voltage drop constraints, see Section 2.2.7) and the rating of the corresponding feeder's protective device (RCCB). Two sets are going to be defined: $F^{\text{PM}} \subseteq F$ for feeders with PM and $F^{\overline{\text{PM}}} \subseteq F$ for feeders without PM. Once again, note that:

$$F^{\text{PM}} \cup F^{\overline{\text{PM}}} = F$$

and

$$F^{\text{PM}} \cap F^{\overline{\text{PM}}} = \emptyset$$

5.2 GENERAL CONSTRAINTS

LOADS AND CHANNELS

Let us now look at the constraints' summary in Sections 4.1.3, 4.2.2 and 4.3.2. The first two:

1. each optional load can only be allocated to its predefined box, and
2. each optional load can only be allocated to channels that supply the necessary current rating and that are not already taken by standard loads,

are already considered in the sets S_l and L_s, previously defined. Constraints three and four:

3. each channel cannot supply more than one load, and

4. each load must be allocated to one channel,

can be stated (respectively) as:

$$\sum_{l \in L_s} x_{l,s} \leq 1,\ \forall s \in S \tag{5.1}$$

$$\sum_{s \in S_l} x_{l,s} = 1,\ \forall l \in L \tag{5.2}$$

$$x_{l,s} \in \{0, 1\} \tag{5.3}$$

The modeling of constraint 5, associated with applicable limits, depends on the optimization target considered. Namely, when changing of cables is allowed (weight optimization), the applicable limits are dependent on these choices. For this reason, this constraint is examined later.

3-phase loads

In Section 2.2.2, a possible restriction in the allocation of three-phase loads was described. This is addressed by constraint number 6:

6. the three phases of a 3-phase load must be connected to three consecutive channels, beginning in phase A.

For the purpose of modeling this constraint, let us consider that the loads can be written as l_i, with $i = 1, \ldots, N_L$. As it was explained in the description of the system (Section 2.2.2), three-phase loads can be considered as three single-phase consecutive loads. Therefore, if l_i is the first resulting load of a three-phase load, l_{i+1} and l_{i+2} complete the group of the three single-phase loads. Note that set L already considers these groups of three loads, since the number of loads N_L did not change. Let $L_T \subseteq L$ be the subset of loads starting a group of a three-phase load. Following this definition, note that S_{l_i} is the set of channels where load l_i can be connected. For simplicity reasons, this set will be referred to merely as S_i.

We can do the same for the channels, i.e. they can be ordered and numbered as s_j, with $j = 1, \ldots, N_S$. The constraint can then be written as:

$$-1 \leq \sum_{s_j \in S_{i+1}} j x_{l_{i+1}, s_j} - \sum_{s_j \in S_i} j x_{l_i, s_j} \leq 1,\ l_i \in L_T \tag{5.4a}$$

$$-1 \leq \sum_{s_j \in S_{i+2}} j x_{l_{i+2}, s_j} - \sum_{s_j \in S_{i+1}} j x_{l_{i+1}, s_j} \leq 1,\ l_i \in L_T \tag{5.4b}$$

Example. To better understand these inequalities, consider the following example:

	A	B	C	A	B	C
	s_1	s_2	s_3	s_4	s_5	s_6
l_1	0	0	0	1	0	0
l_2	0	0	0	0	1	0
l_3	0	0	0	0	0	1

	A	B	C	A	B	C
	s_1	s_2	s_3	s_4	s_5	s_6
l_1	0	0	0	1	0	0
l_2	0	0	0	0	1	0
l_3	0	0	1	0	0	0

	A	B	C	A	B	C
	s_1	s_2	s_3	s_4	s_5	s_6
l_1	0	0	0	1	0	0
l_2	0	0	1	0	0	0
l_3	0	1	0	0	0	0

This shows a situation with three single-phase loads that result from a three-phase load (hence $l_1 \in L_T$) and six available channels. The first case should be the only feasible allocation. Let us verify the constraints:

Allocation 1

$$-1 \leq 5 - 4 \leq 1 \quad \checkmark$$

$$-1 \leq 6 - 5 \leq 1 \quad \checkmark$$

Allocation 2

$$
\begin{aligned}
-1 &\leq 5 - 4 \leq 1 \quad ✓ \\
-1 &\leq 3 - 5 \leq 1 \quad ✗
\end{aligned}
$$

Allocation 3

$$
\begin{aligned}
-1 &\leq 0 - 4 \leq 1 \quad ✗ \\
-1 &\leq 0 - 0 \leq 1 \quad ✓(✗)
\end{aligned}
$$

The calculations for allocations 1 and 2 are straightforward. Allocation 3 appears to satisfy the constraints, when we look at the table. Why are there zeros in the inequalities calculation? The answer comes from the sets S_1, S_2 and S_3 (sets of channels where load l_1, l_2 and l_3 can be allocated, respectively). As discussed in Section 2.2.2, these sets already take into account the electrical phase of each load. S_1 contains only channels supplying phase A, S_2 only phase B channels, and S_3 only phase C channels. Moreover, for the same reason, although loads l_2 and l_3 satisfy the second constraint, they would not satisfy Equation (5.2). Therefore, when used together with the remaining, these constraints properly exclude allocations 2 and 3, making allocation 1 the only feasible solution.

A probably more intuitive method of defining this constraint can be stated as

$$x_{l_{i+1},s_{j+1}} = x_{l_i,s_j}, \quad l_i \in L_T,\ \forall s_j \in S_i \tag{5.5a}$$

$$x_{l_{i+2},s_{j+2}} = x_{l_{i+1},s_{j+1}}, \quad l_i \in L_T,\ \forall s_j \in S_i \tag{5.5b}$$

When load l_i is allocated in channel s_j ($x_{l_i,s_j} = 1$), load l_{i+1} is forced to be connected to channel s_{j+1} and load l_{i+2} to channel s_{j+2}. This also works together with the premise that S_i only contains phase A channels. If l_i is not allocated to channel s_j ($x_{l_i,s_j} = 0$), then it is also guaranteed that loads l_{i+1} and l_{i+2} will not be allocated to the following channels.

Both these two formulations have some advantages and disadvantages. More on this will be covered in Chapter 7.

OPTIONAL CARDS

Some situations allow cards to be added or changed. These cases were identified in Section 2.2.1. The constraint related to optional cards is present in both phase balancing and weight optimizations (these cases do not exist in further allocation possibilities optimization):

7. Only a single type of card must be selected for each card slot (leaving it empty is also an option)

Let $S_k^{\text{opt}} \in S$ be the subset of channels corresponding to the optional card $k = 1, \ldots, N_k$. Each k represents a card that can be chosen or changed. For each optional card k, there are N_{opt_k} different card options. The number of channels $N_{S_k^{\text{opt}}}$ is equal to the total number of channels contained in the possible card choices. Defining $S_k^{\text{opt}_i}$ as the set of slots of the i^{th} card option ($\cup_{i=1}^{N_{\text{opt}_k}} S_k^{\text{opt}_i} = S_k^{\text{opt}}$), the constraints for each of the N_k optional cards, can be stated as:

$$\sum_{l \in L_s} x_{l,s} \leq 1 - \sum_{l \in L_\nu} x_{l,\nu}, \quad i = 1, \ldots, N_{\text{opt}_k},\ \forall s \in S_k^{\text{opt}_i},\ \forall \nu \in S_k^{\text{opt}} \setminus S_k^{\text{opt}_i} \tag{5.6}$$

Consider, for instance, that there are three card options. The inequalities for each optional card can be written as:

$$\sum_{l\in L_s} x_{l,s} \leq 1 - \sum_{l\in L_\nu} x_{l,\nu}, \quad \forall s \in S_k^{\mathrm{opt}_1},\ \forall \nu \in S_k^{\mathrm{opt}_2} \tag{5.7a}$$

$$\sum_{l\in L_s} x_{l,s} \leq 1 - \sum_{l\in L_\nu} x_{l,\nu}, \quad \forall s \in S_k^{\mathrm{opt}_1},\ \forall \nu \in S_k^{\mathrm{opt}_3} \tag{5.7b}$$

$$\sum_{l\in L_s} x_{l,s} \leq 1 - \sum_{l\in L_\nu} x_{l,\nu}, \quad \forall s \in S_k^{\mathrm{opt}_2},\ \forall \nu \in S_k^{\mathrm{opt}_1} \tag{5.7c}$$

$$\sum_{l\in L_s} x_{l,s} \leq 1 - \sum_{l\in L_\nu} x_{l,\nu}, \quad \forall s \in S_k^{\mathrm{opt}_2},\ \forall \nu \in S_k^{\mathrm{opt}_3} \tag{5.7d}$$

$$\sum_{l\in L_s} x_{l,s} \leq 1 - \sum_{l\in L_\nu} x_{l,\nu}, \quad \forall s \in S_k^{\mathrm{opt}_3},\ \forall \nu \in S_k^{\mathrm{opt}_1} \tag{5.7e}$$

$$\sum_{l\in L_s} x_{l,s} \leq 1 - \sum_{l\in L_\nu} x_{l,\nu}, \quad \forall s \in S_k^{\mathrm{opt}_3},\ \forall \nu \in S_k^{\mathrm{opt}_2} \tag{5.7f}$$

Expressions (5.7a) and (5.7b) state that if there is at least one load allocated to a channel belonging to card option 1 ($s \in S_k^{\mathrm{opt}_1}$), there cannot be any load connected to any of the channels belonging to the card option 2 ($\nu \in S_k^{\mathrm{opt}_2}$) or any of the channels belonging to card option 3 ($\nu \in S_k^{\mathrm{opt}_3}$). However, it should be apparent, that there are repeated expressions: (5.7a) is equivalent to (5.7c), (5.7b) to (5.7e), and (5.7d) to (5.7f). To eliminate these repetitions, expression (5.6) can be rewritten as:

$$\sum_{l\in L_s} x_{l,s} \leq 1 - \sum_{l\in L_\nu} x_{l,\nu}, \quad i = 1,\ldots,N_{\mathrm{opt}_k},\ \forall s \in S_k^{\mathrm{opt}_i},\ \forall \nu \in \cup_{j=i+1}^{N_{\mathrm{opt}_k}} S_k^{\mathrm{opt}_j} \tag{5.8}$$

A compact formulation can be given by combining the former expressions. Consider expression (5.7a). It can be written as:

$$\sum_{s\in S_k^{\mathrm{opt}_1}} \sum_{l\in L_s} x_{l,s} \leq N_{S_k^{\mathrm{opt}_1}} \left(1 - \sum_{l\in L_\nu} x_{l,\nu}\right), \quad \forall \nu \in S_k^{\mathrm{opt}_2} \tag{5.9}$$

where $N_{S_k^{\mathrm{opt}_1}}$ is the number of channels belonging to card option 1 (for optional card k). The total expression for the optional card constraint is then given by:

$$\sum_{s\in S_k^{\mathrm{opt}_i}} \sum_{l\in L_s} x_{l,s} \leq N_{S_k^{\mathrm{opt}_i}} \left(1 - \sum_{l\in L_\nu} x_{l,\nu}\right), \quad i = 1,\ldots,N_{\mathrm{opt}_k},\ \forall \nu \in \cup_{j=i+1}^{N_{\mathrm{opt}_k}} S_k^{\mathrm{opt}_j} \tag{5.10}$$

Example. Consider an optional card k with two card options. The first has three channels $S_k^{\mathrm{opt}_1} = \{s_1, s_2, s_3\}$ and the second contains six channels $S_k^{\mathrm{opt}_2} = \{s_4, s_5, s_6, s_7, s_8, s_9\}$. The resulting expressions are

$$\sum_{l\in L_{s_1}} x_{l,s_1} + \sum_{l\in L_{s_2}} x_{l,s_2} + \sum_{l\in L_{s_3}} x_{l,s_3} \leq 3\left(1 - \sum_{l\in L_{s_4}} x_{l,s_4}\right) \tag{5.11a}$$

$$\sum_{l\in L_{s_1}} x_{l,s_1} + \sum_{l\in L_{s_2}} x_{l,s_2} + \sum_{l\in L_{s_3}} x_{l,s_3} \leq 3\left(1 - \sum_{l\in L_{s_5}} x_{l,s_5}\right) \tag{5.11b}$$

$$\ldots$$

$$\sum_{l\in L_{s_1}} x_{l,s_1} + \sum_{l\in L_{s_2}} x_{l,s_2} + \sum_{l\in L_{s_3}} x_{l,s_3} \leq 3\left(1 - \sum_{l\in L_{s_9}} x_{l,s_9}\right) \tag{5.11c}$$

Recall from expression (5.1) that each of the above summations represents a channel with no more than one allocated load. The resulting expressions work as follows: if at least one load is allocated to any of the channels of the second card option, the right-hand term of at least one of the inequalities vanishes, and the left-hand term must be equal to zero. The same is to say: no load can be allocated in a channel belonging to the first card option. If no load is allocated to the second card, all inequalities become the same, and a maximum number of three loads can be allocated to the first card option. Basically, this enables the card to have any amount of loads, from free to completely filled.

This can be verified with the help of two allocation examples:

	A	B	C	A	B	C	A	B	C
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
l_1	0	1	0	0	0	0	0	0	0
l_2	0	0	1	0	0	0	0	0	0

	A	B	C	A	B	C	A	B	C
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
l_1	1	0	0	0	0	0	0	0	0
l_2	0	0	0	0	0	1	0	0	0

The different colors distinguish the channels belonging to the different types of possible card choices Let us consider that all of the channels are available to both loads (the current rating is appropriate and all of them are single-phase loads). The first allocation is allowed, since both loads are connected to the same card option. Hence, a choice of card option has been made. All the inequalities give the same result:

$$2 \leq 3 \quad ✓$$

On the other hand, the second allocation is not possible. The loads are allocated to different card options and consequently no choice for card installation can be made. One of the inequalities (the one containing $\sum_{l \in L_{s_6}} x_{l,s_6}$) correctly eliminates this allocation possibility:

$$1 \leq 0 \quad ✗$$

Finally, another alternative is presented, using an extra set of binary variables $r_{i,k}$:

$$\sum_{s \in S_k^{opt_i}} \sum_{l \in L_s} x_{l,s} \leq N_{S_k^{\mathrm{opt}_i}} r_{i,k}, \quad i = 1, \ldots, N_{\mathrm{opt}_k} \tag{5.12a}$$

$$\sum_{i=1}^{N_{\mathrm{opt}_k}} r_{i,k} \leq 1, \quad k = 1, \ldots, N_K \tag{5.12b}$$

$$r_{i,k} \in \{0, 1\} \tag{5.12c}$$

For each optional card k, no more than one variable $r_{i,k}$ can be equal to 1. If $r_{i,k}$ is equal to 1, loads can be allocated to channels belonging to card option i. For the variables $r_{i,k}$ equal to 0, no load can be allocated to the channels corresponding to those card options. Since all the $r_{i,k}$ can be equal to 0, the possibility of leaving the card slot free is assured.

The three possibilities presented ((5.8), (5.10) and (5.12)) are going to be studied in a bit more detail in Chapter 7, so that a single formulation is chosen for the final implementation.

5.3 Three-phase power unbalance

5.3.1 Constraints

Applicable limits

Before explaining the formulations for the applicable limits, it is important to remember the difference between power limit and applicable limit. Power limit is the maximum power that a device can supply without damage; applicable limit is the value used for the calculation of the constraints, which for various reasons can be different from the power limit.

In the unbalance optimization, unlike the full weight optimization case, the segments are fixed and therefore, the power limits (related to the maximum current a cable can supply) are known *a priori*. Each segment has a power limit of $m_{f,b}$, where once again $f \in F$ is the feeder and $b = 1, \ldots, N_{B_f}$ is the index of the box/segment (see Figure 5.1). The value $m_{f,0}$ describes the power limit of feeder f, which is the power limit of the protection device (RCCB). As discussed in Section 2.2.7, there is normally no reason to consider the rating of the protective device different from the first segment's limit. However, other segments may have different power limits, in order to reduce their cross-section. If so, the applicable limits may differ. Therefore, let B_f^{eq} be the set of segments with the same current limit as the corresponding feeder RCCB's rating, and B_f^{lo} be the set of segments with lower current limit. Hence: $B_f^{\text{eq}} \cup B_f^{\text{lo}} = B_f$, since no segment can have a larger current limit than the RCCB (this would lead to a trip in this device).

The power contribution of the standard loads can be considered with the help of this set of constants: $\Omega^{\phi,\text{o},\text{per},\text{sh}}_{\gamma,f,b}$, $\Omega^{\phi,\text{o},\text{per},\overline{\text{sh}}}_{\gamma,f,b}$, $\Omega^{\phi,\text{o},\text{int},\text{sh}}_{\gamma,f,b}$, $\Omega^{\phi,\text{o},\text{int},\overline{\text{sh}}}_{\gamma,f,b}$, $\Omega^{\phi,\text{m},\text{per},\text{sh}}_{\gamma,f,b}$, $\Omega^{\phi,\text{m},\text{per},\overline{\text{sh}}}_{\gamma,f,b}$, $\Omega^{\phi,\text{m},\text{int},\text{sh}}_{\gamma,f,b}$, $\Omega^{\phi,\text{m},\text{int},\overline{\text{sh}}}_{\gamma,f,b}$, where Ω is the sum of the powers of the standard loads, considering the different flight phases γ, the distinct feeders f and boxes $b = 1, \ldots, N_{B_f}$, the electrical phase ϕ, the type of power (o for operational and m for maximum), the type of operation *per*(manent) and *int*(ermittent), and sheddable and non-sheddable loads. The same applies here, $b = 0$ is the power contribution of all the standard loads supplied by a given feeder, for example,

$$\Omega^{\phi,\text{o},\text{per},\text{sh}}_{\gamma,f,0} = \sum_{b=1}^{N_{B_f}} \Omega^{\phi,\text{o},\text{per},\text{sh}}_{\gamma,f,b}.$$

Moreover, the power of the standard loads that goes through segment i is given by

$$\sum_{b=i}^{N_{B_f}} \Omega^{\phi,\text{o},\text{per},\text{sh}}_{\gamma,f,b},$$

since all the downstream boxes contribute to the power of a given segment.

The applicable limits can have different combinations regarding the types of load power. Also, they must consider the distinction between feeders with and without PM, and the cases where the current limit of the segment is lower than the rating of the feeder's protective device.

Example. According to the example given in Section 2.2.7, the applicable limits for feeders with PM were given by the following table:

ΔV Constraint	Applicable Limit	Maximum Power	Operational Power	Permanent	Intermittent	Non-Sheddable	Sheddable
-	87%	X	-	X	X	X	-
X	87%	X	-	X	X	X	X
-	200%	X	-	X	X	X	X

The constraints would then be written as

$$0.87 \cdot m_{f,0} \geq \sum_{s \in S^{\phi}_{f,0}} \sum_{l \in L_{\overline{\text{sh}}}} x_{l,s} \left(p^{\text{m,per}}_{l,\gamma} + p^{\text{m,int}}_{l,\gamma} \right) + \Omega^{\phi,\text{m},\text{per},\overline{\text{sh}}}_{\gamma,f,0} + \Omega^{\phi,\text{m},\text{int},\overline{\text{sh}}}_{\gamma,f,0},$$

$$\forall f \in F^{\text{PM}},\ \gamma = 1, \ldots, N_{\Gamma},\ \phi = 1,2,3 \tag{5.13a}$$

$$2 \cdot m_{f,0} \geq \sum_{s \in S^{\phi}_{f,0}} \sum_{l \in L_{\text{sh}} \cup L_{\overline{\text{sh}}}} x_{l,s} \left(p^{\text{m,per}}_{l,\gamma} + p^{\text{m,int}}_{l,\gamma} \right) + \Omega^{\phi,\text{m},\text{per},\text{sh}}_{\gamma,f,0} + \Omega^{\phi,\text{m},\text{per},\overline{\text{sh}}}_{\gamma,f,0} + \Omega^{\phi,\text{m},\text{int},\text{sh}}_{\gamma,f,0} + \Omega^{\phi,\text{m},\text{int},\overline{\text{sh}}}_{\gamma,f,0},$$

$$\forall f \in F^{\text{PM}},\ \gamma = 1, \ldots, N_{\Gamma},\ \phi = 1,2,3 \tag{5.13b}$$

These are the applicable limits calculated for the total load consumption on the feeder, which in a normal situation will be the same as the ones calculated for the first segment.

For each of the segments, according to the same example, the following constraints would apply:

$$0.87 \cdot m_{f,b} \geq \sum_{i=b}^{N_{B_f}} \sum_{s \in S^{\phi}_{f,i}} \sum_{l \in L_{\overline{\text{sh}}}} x_{l,s} \left(p^{\text{m,per}}_{l,\gamma} + p^{\text{m,int}}_{l,\gamma}\right) + \Omega^{\phi,\text{m,per},\overline{\text{sh}}}_{\gamma,f,i} + \Omega^{\phi,\text{m,int},\overline{\text{sh}}}_{\gamma,f,i},$$
$$\forall f \in F^{\text{PM}},\ \forall b \in B^{\text{eq}}_f,\ \gamma = 1,\dots,N_\Gamma,\ \phi = 1,2,3 \tag{5.14a}$$

$$2 \cdot m_{f,b} \geq \sum_{i=b}^{N_{B_f}} \sum_{s \in S^{\phi}_{f,i}} \sum_{l \in L_{\text{sh}} \cup L_{\overline{\text{sh}}}} x_{l,s} \left(p^{\text{m,per}}_{l,\gamma} + p^{\text{m,int}}_{l,\gamma}\right) + \Omega^{\phi,\text{m,per,sh}}_{\gamma,f,i} + \Omega^{\phi,\text{m,per},\overline{\text{sh}}}_{\gamma,f,i} + \Omega^{\phi,\text{m,int,sh}}_{\gamma,f,i} + \Omega^{\phi,\text{m,int},\overline{\text{sh}}}_{\gamma,f,i},$$
$$\forall f \in F^{\text{PM}},\ \forall b \in B^{\text{eq}}_f,\ \gamma = 1,\dots,N_\Gamma,\ \phi = 1,2,3 \tag{5.14b}$$

$$0.87 \cdot m_{f,b} \geq \sum_{i=b}^{N_{B_f}} \sum_{s \in S^{\phi}_{f,i}} \sum_{l \in L_{\text{sh}} \cup L_{\overline{\text{sh}}}} x_{l,s} \left(p^{\text{m,per}}_{l,\gamma} + p^{\text{m,int}}_{l,\gamma}\right) + \Omega^{\phi,\text{m,per,sh}}_{\gamma,f,i} + \Omega^{\phi,\text{m,per},\overline{\text{sh}}}_{\gamma,f,i} + \Omega^{\phi,\text{m,int,sh}}_{\gamma,f,i} + \Omega^{\phi,\text{m,int},\overline{\text{sh}}}_{\gamma,f,i},$$
$$\forall f \in F^{\text{PM}},\ \forall b \in B^{\text{lo}}_f,\ \gamma = 1,\dots,N_\Gamma,\ \phi = 1,2,3 \tag{5.14c}$$

Note that the last constraint applies to segments with lower current limit than the protection device's rating (second line of the table), while the others apply to segments with the same current limit as the protection device's rating (first and third line of the table). For all of them, the power consumptions on a cable segment include all power consumptions of the boxes located downstream (summation $b = i, \dots, N_{B_f}$).

According to the same example, for feeders and segments without PM, only the limit with no over-installation would be calculated, considering all loads as non-sheddable:

$$0.87 \cdot m_{f,0} \geq \sum_{s \in S^{\phi}_{f,0}} \sum_{l \in L_{\text{sh}} \cup L_{\overline{\text{sh}}}} x_{l,s} \left(p^{\text{m,per}}_{l,\gamma} + p^{\text{m,int}}_{l,\gamma}\right) + \Omega^{\phi,\text{m,per},\overline{\text{sh}}}_{\gamma,f,0} + \Omega^{\phi,\text{m,int},\overline{\text{sh}}}_{\gamma,f,0},$$
$$\forall f \in F^{\overline{\text{PM}}},\ \gamma = 1,\dots,N_\Gamma,\ \phi = 1,2,3 \tag{5.15a}$$

$$0.87 \cdot m_{f,b} \geq \sum_{i=b}^{N_{B_f}} \sum_{s \in S^{\phi}_{f,i}} \sum_{l \in L_{\text{sh}} \cup L_{\overline{\text{sh}}}} x_{l,s} \left(p^{\text{m,per}}_{l,\gamma} + p^{\text{m,int}}_{l,\gamma}\right) + \Omega^{\phi,\text{m,per,sh}}_{\gamma,f,i} + \Omega^{\phi,\text{m,per},\overline{\text{sh}}}_{\gamma,f,i} + \Omega^{\phi,\text{m,int,sh}}_{\gamma,f,i} + \Omega^{\phi,\text{m,int},\overline{\text{sh}}}_{\gamma,f,i},$$
$$\forall f \in F^{\overline{\text{PM}}},\ \forall b \in B^{\text{eq}}_f,\ \gamma = 1,\dots,N_\Gamma,\ \phi = 1,2,3 \tag{5.15b}$$

$$0.87 \cdot m_{f,b} \geq \sum_{i=b}^{N_{B_f}} \sum_{s \in S^{\phi}_{f,i}} \sum_{l \in L_{\text{sh}} \cup L_{\overline{\text{sh}}}} x_{l,s} \left(p^{\text{m,per}}_{l,\gamma} + p^{\text{m,int}}_{l,\gamma}\right) + \Omega^{\phi,\text{m,per,sh}}_{\gamma,f,i} + \Omega^{\phi,\text{m,per},\overline{\text{sh}}}_{\gamma,f,i} + \Omega^{\phi,\text{m,int,sh}}_{\gamma,f,i} + \Omega^{\phi,\text{m,int},\overline{\text{sh}}}_{\gamma,f,i},$$
$$\forall f \in F^{\overline{\text{PM}}},\ \forall b \in B^{\text{lo}}_f,\ \gamma = 1,\dots,N_\Gamma,\ \phi = 1,2,3 \tag{5.15c}$$

Again, for segments with lower rating, all power contributions are considered. These constraints serve merely as an example. Operational power can also be present in some of the constraints and different combinations can occur. The main point here is that they depend on the type of feeder (with or without PM) and on the relation between the current limit (power limit) of the segment and the rating of the protection device.

5.3.2 TARGETS

For the optimization targets, only operational power is considered (the reasons are given in Section 4.1.2), with both sheddable and non-sheddable loads taken into consideration. The optimization is done on the feeder level. As explained in Section 4.1.1, the unbalance will be calculated by Equation (4.9). For simplicity reasons, we define:

$$P^{\phi}_{\gamma,f} =: \sum_{s \in S^{\phi}_{f,0}} \sum_{l \in L_{\text{sh}} \cup L_{\overline{\text{sh}}}} x_{l,s} p^{\text{o,per}}_{l,\gamma} + \Omega^{\phi,\text{o,per,sh}}_{\gamma,f} + \Omega^{\phi,\text{o,per},\overline{\text{sh}}}_{\gamma,f} \tag{5.16}$$

which stands for the relevant power consumption (in terms of the objective functions) in a given feeder f, during a given flight phase γ and on electrical phase $\phi = 1, 2, 3$. The power difference between two electrical phases is then given by:

$$U_{\gamma,f}^{\phi_1,\phi_2} =: P_{\gamma,f}^{\phi_1} - P_{\gamma,f}^{\phi_2} \tag{5.17}$$

Finally, the unbalance for a given flight phase γ and a certain feeder f can be written as:

$$U_{\gamma,f} =: \max_{\phi_1,\phi_2} U_{\gamma,f}^{\phi_1,\phi_2} \tag{5.18}$$

Using these expressions, the following targets are defined (see Section 4.1.2):

Target 1. Minimize the maximum unbalance among all feeders and all flight phases.

$$\begin{aligned} \underset{x=(x_{l,s})}{\text{minimize}} \quad u(x) &= \max_{\gamma,f} U_{\gamma,f} \\ &=: U \end{aligned} \tag{5.19}$$

$$\begin{aligned} \text{subject to} \quad & (5.1) - (5.3) \\ & \text{either } (5.4) \text{ or } (5.5) \\ & \text{either } (5.8),\ (5.10) \text{ or } (5.12) \\ & (5.13) - (5.15) \\ & U \geq U_{\gamma,f}, \quad f \in F,\ \gamma = 1, \ldots, N_\Gamma & \text{(5.19a)} \\ & U_{\gamma,f} \geq U_{\gamma,f}^{\phi_1,\phi_2}, \quad f \in F,\ \gamma = 1, \ldots, N_\Gamma,\ \forall \phi_1, \phi_2 & \text{(5.19b)} \end{aligned}$$

Although easier to understand with these two steps (first define the maximum difference between powers, $U_{\gamma,f}$, for each flight phase and each feeder, and then calculate the maximum of all these values), this formulation uses unnecessary variables. A more efficient approach (in terms of variables), is given by:

$$\begin{aligned} \underset{x=(x_{l,s})}{\text{minimize}} \quad u(x) &= \max_{\gamma,f,\phi_1,\phi_2} U_{\gamma,f}^{\phi_1,\phi_2} \\ &=: U \end{aligned} \tag{5.20}$$

$$\begin{aligned} \text{subject to} \quad & (5.1) - (5.3) \\ & \text{either } (5.4) \text{ or } (5.5) \\ & \text{either } (5.8),\ (5.10) \text{ or } (5.12) \\ & (5.13) - (5.15) \\ & U \geq U_{\gamma,f}^{\phi_1,\phi_2}, \quad f \in F,\ \gamma = 1, \ldots, N_\Gamma,\ \forall \phi_1, \phi_2 & \text{(5.20a)} \end{aligned}$$

This formulation has $N_L \times N_S$ load-channel variables ($x_{i,j}$) and an extra variable U. The different $U_{\gamma,f}^{\phi_1,\phi_2}$ can be written as a function of the $x_{i,j}$ variables. Hence, the total number of variables is equal to $N_L \times N_S + 1$.

Target 2. Minimize the mean unbalance among all feeders and all flight phases.

$$\underset{x=(x_{l,s})}{\text{minimize}} \quad u(x) = \frac{1}{N_F \cdot N_\Gamma} \sum_{f \in F} \sum_{\gamma=1}^{N_\Gamma} U_{\gamma,f} \tag{5.21}$$

$$\begin{aligned} \text{subject to} \quad & (5.1) - (5.3) \\ & \text{either } (5.4) \text{ or } (5.5) \\ & \text{either } (5.8),\ (5.10) \text{ or } (5.12) \\ & (5.13) - (5.15) \\ & U_{\gamma,f} \geq U_{\gamma,f}^{\phi_1,\phi_2}, \quad f \in F,\ \gamma = 1, \ldots, N_\Gamma,\ \forall \phi_1, \phi_2 & \text{(5.21a)} \end{aligned}$$

The normalizing factor in the objective function has no influence on the allocation choices. However, it can be useful when reading the result. The formulation has again $N_L \times N_S$ load-channel variables, but now it uses $N_\Gamma \times N_F$ extra variables $(U_{\gamma,f})$. A similar objective function can also be used to assign difference importances to the flight phases:

$$\underset{x=(x_{l,s})}{\text{minimize}} \quad u(x) = \frac{1}{N_F \cdot \sum_{\gamma=1}^{N_\Gamma} a_\gamma} \sum_{f \in F} \sum_{\gamma=1}^{N_\Gamma} a_\gamma U_{\gamma,f} \tag{5.22}$$

where a_γ is a real valued positive constant for each flight phase γ. The larger the value, the more importance is given to a certain flight phase. Once again, the constant multiplying the summation of the unbalance over all flight phases is only used to simplify the reading of the result. This can be also regarded as a case of multi-objective optimization, more precisely the weighted sum of objectives explained in Section 3.5.2.

Target 3. Minimize the maximum unbalance among all feeders, considering the powers on each electrical phase as the maximum among all flight phases.

$$\begin{aligned} \underset{x=(x_{l,s})}{\text{minimize}} \quad u(x) &= \max_{f,\phi_1,\phi_2} \left(\max_{\gamma_1} P^{\phi_1}_{\gamma_1,f} - \max_{\gamma_2} P^{\phi_2}_{\gamma_2,f} \right) \\ &=: \max_{f,\phi_1,\phi_2} \left(P^{\phi_1}_f - P^{\phi_2}_f \right) \\ &=: U \end{aligned} \tag{5.23}$$

$$\begin{aligned} \text{subject to} \quad & (5.1) - (5.3) \\ & \text{either } (5.4) \text{ or } (5.5) \\ & \text{either } (5.8),\ (5.10) \text{ or } (5.12) \\ & (5.13) - (5.15) \\ & U \geq P^{\phi_1}_f - P^{\phi_2}_f, \quad \forall f \in F, \phi_1, \phi_2 && (5.23a) \\ & P^{\phi}_f \geq P^{\phi}_{\gamma,f}, \quad \forall f \in F,\ \forall \phi && (5.23b) \\ & P^{\phi}_f \leq P^{\phi}_{\gamma,f} + M(1 - z^{\phi}_{\gamma,f}), \quad \forall f \in F,\ \forall \phi && (5.23c) \\ & \sum_{\gamma=1}^{\Gamma} z^{\phi}_{\gamma,f} = 1 && (5.23d) \\ & z^{\phi}_{\gamma,f} \in \{0, 1\} && (5.23e) \end{aligned}$$

where M is a sufficiently large constant. Constraints (5.23b)-(5.23d) are used to select the maximum unbalance over each electrical phase. Inequality (5.23b) is the usual way for obtaining a maximum: an extra variable is created, with the constraint that it must be greater or equal than all of the values $P^{\phi}_{\gamma,f}$ considered for the maximum. The problem here, when compared to previous situations, is that we are not minimizing variables P^{ϕ}_f. Hence, there is no guarantee that the lowest possible value for each P^{ϕ}_f will be selected. To force one of the $P^{\phi}_{\gamma,f}$ to be chosen, inequalities (5.23c) and (5.23d) are added. With the big M constraint, P^{ϕ}_f must be lower or equal to exactly one of the values $P^{\phi}_{\gamma,f}$. This will be exactly the maximum value, since P^{ϕ}_f must be greater or equal than all of them.

5.4 WEIGHT

5.4.1 CONSTRAINTS

CABLE SELECTION

For the weight optimization, every cable can be chosen. There is a fixed number of cable possibilities N_C, and it will be assumed that these have the same power limits as the possibilities for the protection device. We define the binary variables $y_{f,b,c}$, where $f \in F$ is the feeder, $b = 1, \ldots, N_{B_F}$ the number of the box or feeder's segment (see Figure 5.1) and $c = 1, \ldots, N_C$ the type of cable. The value $b = 0$ is used as the variable for the protection device. Let also m_c be the power limit associated with cable type c. Regarding the cables, as discussed in Section 2.2.7, these limits can result from voltage drop specifications and can be lower than the actual power they can supply without damage. But for the optimization this information is not needed. Since only one type of cable/protection device must be selected (and selecting no cable/protection device is not an option), the following constraints shall be verified for the protective device and each segment of the feeder:

$$\sum_{c=1}^{C} y_{f,b,c} = 1, \quad \forall f \in F,\ b = 0, \ldots, N_{B_f} \tag{5.24a}$$

$$y_{f,b,c} \in \{0, 1\} \tag{5.24b}$$

APPLICABLE LIMITS

It will be considered that the power limits of the segments can never be larger than the power limit of the protection device, since this would lead to situations where the cables could not be protected. For this matter, we can define an auxiliary variable:

$$d_{f,b} = \sum_{c_1=1}^{C} c_1 y_{f,0,c_1} - \sum_{c_2=1}^{C} c_2 y_{f,b,c_2}, \quad \forall f \in F,\ \forall b = 1, \ldots, N_{B_f} \tag{5.25}$$

If we consider increasing power limit for the cable types c, these variables $d_{f,b}$ take a value larger or equal to 0. Note that $d_{f,b} = 0$ when the segment chosen has the same power limit as the protection device, and $d_{f,b} \geq 1$, when the segment chosen has a lower power limit than the protective device. Moreover, $d_{f,b} \leq C - 1$. As it was previously discussed, we are interested in discriminating only these two situations, so the different values greater than 1 that $d_{f,b}$ can take are not important for our analysis. With that in mind, we can define a set of binary variables $z_{f,b}$ that are set to 0 when segment b has the same power limit as its corresponding protection device, and set to 1 when the power limit is lower. This can be fulfilled applying the following conditions:

$$z_{f,b} \leq d_{f,b} \tag{5.26a}$$

$$z_{f,b} \geq \frac{1}{C-1} d_{f,b} \tag{5.26b}$$

$$z_{f,b} \in \{0, 1\} \tag{5.26c}$$

With this, if the segment has the same power limit as the protection device, that is, $d_{f,b} = 0$, the first equation will force $z_{f,b} = 0$. When the segment has lower power limit than the protective device, $1 \leq d_{f,b} \leq C - 1$ and the second equation forces $z_{f,b} = 1$. Due to the different applicable limits, depending on the relation between the segment and the protective device power limits, the constraints vary with the choice of cable type. This can be modeled using the variables $z_{f,b}$ priorly defined, together with the help of a large value constant M (this procedure was explained in Chapter 3). As an example, constraints (5.14) become:

$$
\begin{aligned}
0.87 \cdot \sum_{c=1}^{C} y_{f,b,c} m_c \geq \\
\sum_{i=b}^{N_{B_f}} \sum_{s \in S_{f,i}^{\phi}} \sum_{l \in L_{\overline{\mathrm{sh}}}} x_{l,s} \left(p_{l,\gamma}^{\mathrm{m,per}} + p_{l,\gamma}^{\mathrm{m,int}} \right) + \Omega_{\gamma,f,i}^{\phi,\mathrm{m,per},\overline{\mathrm{sh}}} + \Omega_{\gamma,f,i}^{\phi,\mathrm{m,int},\overline{\mathrm{sh}}} - z_{f,b} M, \\
\forall f \in F^{\mathrm{PM}},\ \forall b \in B_f,\ \gamma = 1, \ldots, N_{\Gamma},\ \phi = 1,2,3
\end{aligned} \tag{5.27a}
$$

$$
\begin{aligned}
2 \cdot \sum_{c=1}^{C} y_{f,b,c} m_c \geq \\
\sum_{i=b}^{N_{B_f}} \sum_{s \in S_{f,i}^{\phi}} \sum_{l \in L_{\mathrm{sh}} \cup L_{\overline{\mathrm{sh}}}} x_{l,s} \left(p_{l,\gamma}^{\mathrm{m,per}} + p_{l,\gamma}^{\mathrm{m,int}} \right) + \Omega_{\gamma,f,i}^{\phi,\mathrm{m,per,sh}} + \Omega_{\gamma,f,i}^{\phi,\mathrm{m,per},\overline{\mathrm{sh}}} + \Omega_{\gamma,f,i}^{\phi,\mathrm{m,int,sh}} + \Omega_{\gamma,f,i}^{\phi,\mathrm{m,int},\overline{\mathrm{sh}}} - z_{f,b} M, \\
\forall f \in F^{\mathrm{PM}},\ \forall b \in B_f,\ \gamma = 1, \ldots, N_{\Gamma},\ \phi = 1,2,3
\end{aligned} \tag{5.27b}
$$

$$
\begin{aligned}
0.87 \cdot \sum_{c=1}^{C} y_{f,b,c} m_c \geq \\
\sum_{i=b}^{N_{B_f}} \sum_{s \in S_{f,i}^{\phi}} \sum_{l \in L_{\mathrm{sh}} \cup L_{\overline{\mathrm{sh}}}} x_{l,s} \left(p_{l,\gamma}^{\mathrm{m,per}} + p_{l,\gamma}^{\mathrm{m,int}} \right) + \Omega_{\gamma,f,i}^{\phi,\mathrm{m,per,sh}} + \Omega_{\gamma,f,i}^{\phi,\mathrm{m,per},\overline{\mathrm{sh}}} + \Omega_{\gamma,f,i}^{\phi,\mathrm{m,int,sh}} + \Omega_{\gamma,f,i}^{\phi,\mathrm{m,int},\overline{\mathrm{sh}}} - \left(1 - z_{f,b}\right) M, \\
\forall f \in F^{\mathrm{PM}},\ \forall b \in B_f,\ \gamma = 1, \ldots, N_{\Gamma},\ \phi = 1,2,3
\end{aligned} \tag{5.27c}
$$

The first two inequalities apply when the cable selected for segment b has the same power limit as the protection device ($z_{f,b} = 0$) and the last is selected when the cable segment has lower power limit than the protection device ($z_{f,b} = 1$). This is exactly the desired behavior, according to the table of applicable limits.

OPTIONAL CARDS

The weight resulting from the card selection must also be considered. When the constraints related to optional cards were defined, three different models were developed. The third one (expressions (5.12)) is very useful for this purpose, since each variable $r_{i,k}$ is either 0 or 1 if card option i is selected or not for card slot k, respectively. Hence, these variables can be multiplied by the corresponding card weights in the objective function.

5.4.2 TARGETS

Let w_c^{cable} be the weight associated with cable type c and w_i^{card} the weight associated with card type i.

Target 1. Minimize the weight of the system considering cables and cards

$$\underset{x=(x_{l,s})}{\text{minimize}} \quad w(x) = \sum_{f\in F}\sum_{b=1}^{N_{B_f}}\sum_{c=1}^{C} w_c^{\text{cable}} y_{b,f,c} + \sum_{k=1}^{N_K}\sum_{i=1}^{N_{\text{opt}_k}} w_i^{\text{card}} r_{i,k} \tag{5.28}$$

$$\begin{aligned} \text{subject to} \quad & (5.1) - (5.3) \\ & \text{either } (5.4) \text{ or } (5.5) \\ & (5.12) \\ & (5.24) - (5.27) \end{aligned}$$

Target 2. Minimize the weight of the system considering only cards

$$\underset{x=(x_{l,s})}{\text{minimize}} \quad w(X) = \sum_{i=1}^{N_{\text{opt}_k}} w_i^{\text{card}} r_{i,k} \tag{5.29}$$

$$\begin{aligned} \text{subject to} \quad & (5.1) - (5.3) \\ & \text{either } (5.4) \text{ or } (5.5) \\ & (5.12) \\ & (5.13) - (5.15) \end{aligned}$$

Notice that in this case, the applicable limits with fixed cables are used.

5.5 FURTHER ALLOCATION POSSIBILITIES

To optimize further allocation possibilities, the number of possible ratings for each channel must be considered. Let us define h_s as the number of different ratings that channel s can supply. This is enough when no restriction to the connection of three-phase loads exists. In the cases where three-phase loads have to be connected in consecutive channels starting on phase A, let $S_T \in S$ be the first of these three consecutive channels (the one that supplies electrical phase A) and h_s^T be the number of different ratings common to all three consecutive channels starting on $s \in S_T$.

Example. Consider the following card:

	s_1	s_2	s_3	s_4	s_5	s_6
LRM AC-6	A	B	C	A	B	C
	5/10/15	10/15	5/10/15	2/4/5	1/2/4/5	2/4/5
h_s	3	2	3	3	4	3
h_s^T	2			3		

Looking to the first channel s_1, it is possible to see that it can supply three different ratings: 5, 10 and 15 A. Thus, $h_{s_1} = 3$. However, the group s_1, s_2 and s_3, is only able to supply three-phase loads with current ratings of either 10 or 15 A, since s_2 cannot supply 5 A. Therefore, $h_{s_1}^T = 2$.

5.5.1 TARGETS

From the previous example, if a load is connected to channel s_3, for instance, it occupies $h_{s_3} = 3$ possible connections for single-phase loads. If connected to channel s_2 it only occupies $h_{s_2} = 2$ possible connections, making this solution better, if further loads must be allocated. However, if any of them is taken, no three-phase load can be connected to the first group of channels. Therefore, if a load is connected to either s_1, s_2 or s_3, it occupies $h^T_{s_1} = 2$ possible connections for three-phase loads.

For the applicable limits, this situation is similar to the three-phase balancing optimization, since all the cables are considered fixed.

Target 1. Minimize the number of used ratings considering single-phase and three-phase loads.

$$\underset{x=(x_{l,s})}{\text{minimize}} \quad g(x) = \sum_{s\in S}\sum_{l\in L_s} h_s x_{l,s} + \sum_{s_j\in S_T}\sum_{l\in L_{s_j}} h^T_{s_j} t_{s_j} \tag{5.30}$$

$$\begin{aligned}
\text{subject to} \quad & (5.1) - (5.3) \\
& \text{either } (5.4) \text{ or } (5.5) \\
& (5.13) - (5.15) \\
& t_{s_j} \geq \sum_{l\in L_{s_j}} x_{l,s_j} && \text{(5.30a)} \\
& t_{s_j} \geq \sum_{l\in L_{s_{j+1}}} x_{l,s_{j+1}} && \text{(5.30b)} \\
& t_{s_j} \geq \sum_{l\in L_{s_{j+2}}} x_{l,s_{j+2}} && \text{(5.30c)} \\
& t_{s_j} \in \{0,1\} && \text{(5.30d)}
\end{aligned}$$

In the objective function, the summation over variables $x_{l,s}$ deals with the occupation of possibilities to allocate single-phase loads. The idea here is to minimize this occupation. Summation over variables t_{s_j} has the same purpose, but for three-phase loads. Here, s_j is a channel supplying electrical phase A. So channels s_{j+1} and s_{j+2} supply electrical phase B and electrical phase C, respectively. If any of them is taken, variable t_{s_j} is set to 1. This is the role of constraints (5.30a)-(5.30c).

5.6 MULTI-OBJECTIVE PREEMPTIVE OPTIMIZATION

In Section 3.5.1, we looked at the possibility of optimizing the system according to multiple targets. One of the solutions presented, preemptive optimization, is based on the assumption that optimizations can be performed considering one objective at a time, with a specified priority. This is done by including the previous objective functions as constraints of the current optimization (see (3.20b)).

Example. Consider the situation where an optimization in phase balancing is performed, using target 1. As shown before, the result can contain some situations where unnecessary cards are used. To eliminate these cards, weight optimization can be done, using target 2. The formulation to perform this optimization, with a fixed optimal value U^* for the phase balancing, is given by:

$$
\begin{aligned}
\underset{x=(x_{l,s})}{\text{minimize}} \quad & w(X) = \sum_{i=1}^{N_{\text{opt}_k}} w_i^{\text{card}} r_{i,k} && (5.31) \\
\text{subject to} \quad & (5.1) - (5.2) \\
& \text{either } (5.4) \text{ or } (5.5) \\
& (5.12) \\
& (5.13) - (5.15) \\
& x_{l,s} \in \{0,1\} && (5.31a) \\
& U \geq U_{\gamma,f}^{\phi_1,\phi_2}, \quad f \in F,\ \gamma = 1,\ldots,N_\Gamma,\ \forall \phi_1, \phi_2 && (5.31b) \\
& U \leq U^* && (5.31c)
\end{aligned}
$$

Inequality (5.31c) comes from the objective function of the previous optimization and is responsible for keeping the value of the unbalance within the optimal value. As a result, this optimization will give the lowest card weight possible for the given optimal unbalance value. Observe that this optimization only gives a different allocation scheme if there is more than one possibility with the same value of unbalance. Expression (5.31b) is also necessary, since it is a constraint from the previous optimization.

From this example, we conclude that when preemptive optimization is done, all the constraints from previous optimizations shall be added, as well as the constraints that guarantee the preservation of the previous optimal values.

5.7 DC OPTIMIZATION

Before finishing this chapter, it is worth addressing DC optimization briefly. During the definition of the targets, it was mentioned that three-phase balancing is not a DC target, since it only involves AC power. The same applies for all considerations regarding three-phase loads. For instance, when performing DC optimization, the target for the allocation possibilities does not contain the second term. All of the above definitions containing electrical phases ϕ are replaced by a single quantity. As a result, the number of applicable limits is reduced by a factor of three.

CHAPTER 6

MATLAB® Implementation

MATLAB® was the chosen environment to develop the tool. This decision was made taking into consideration the available software in the company. MATLAB® is an high-level language and interactive environment for numerical computation, visualization and programming. It is particularly efficient in dealing with matrices, as the name indicates (*Mat*rix *Lab*oratory). Until 2014, there was no implemented software to deal with Mixed-Integer Linear Programming problems, only LP problems. However, version 2014a (released in March 2014) introduced the function `intlinprog`. In this section, we first look into this function in more detail, focusing on the input and output parameters.

The syntax and description are given by [22]:

Syntax:

```
x = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)
```

Description:

$$\min_x f^{\mathrm{T}} x \text{ subject to } \begin{cases} x(\text{intcon}) \text{ are integers} \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases}$$

f, x, intcon, b, beq, lb, and ub are vectors, and A and Aeq are matrices.

From now on, MATLAB® variables will be written in a similar style as the syntax given above. To represent vector and matrix indexes, the notation from MATLAB® code is used: `x(i)` is the i^{th} element of vector x, `A(i,j)` is the value of the i^{th} line, j^{th} column of matrix A. The notation `A(i,:)` refers to line i inside matrix A (":" means all elements, and can also be applied to the line index). Finally, `A(i1:i2,:)` represents the submatrix of A composed by all the lines in the interval $[i_1, i_2]$. This is similar in the case of vectors.

6.1 GENERAL EXAMPLE

It is probably easier to understand the implementation in MATLAB® with an example. Consider the following problem, based on the general formulation given in Section 3.4:

$$\text{minimize or maximize} \quad f_1x_1 + f_2x_2 + \cdots + f_nx_n \tag{6.1}$$

$$
\begin{aligned}
\text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 && (6.1\text{d})\\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq b_2 && (6.1\text{e})\\
& a_{31}x_1 + a_{32}x_2 + \cdots + a_{3n}x_n = b_3 && (6.1\text{f})\\
& \ldots \\
& x_j \geq 0,\ \forall j = 1, \ldots, n && (6.1\text{g})\\
& x_2, x_4 \text{ integer} && (6.1\text{h})\\
& x_1 \in \{0, 1\}. && (6.1\text{i})
\end{aligned}
$$

In the form for `intlinprog`, vector `x` is the output and it represents the variables, hence

$$\mathtt{x} = \begin{pmatrix} x_1 & x_2 & \ldots & x_n \end{pmatrix}.$$

The coefficients of the linear objective function are described by vector `f`:

$$\mathtt{f} = \begin{pmatrix} f_1 & f_2 & \ldots & f_n \end{pmatrix},$$

where f_i is the coefficient multiplying variable x_i, as given in the example.

To describe the constraints first note that, according to the `intlinprog` form, they should be divided into linear inequalities, linear equalities, and bounds. The first constraint is a linear inequality, which is already in the necessary form, with

$$\mathtt{A} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \end{pmatrix}$$

and

$$\mathtt{b} = \begin{pmatrix} b_1 \end{pmatrix}.$$

The second constraint is not formulated in the same way. However, it is easy to specify it in the necessary form:

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq b_2 \Leftrightarrow -a_{21}x_1 - a_{22}x_2 - \cdots - a_{2n}x_n \leq -b_2,$$

therefore, matrix A becomes

$$\mathtt{A} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ -a_{21} & -a_{22} & \ldots & -a_{2n} \end{pmatrix},$$

and

$$\mathtt{b} = \begin{pmatrix} b_1 \\ -b_2 \end{pmatrix}.$$

The linear equality is written in the same form:

$$\mathtt{Aeq} = \begin{pmatrix} a_{31} & a_{32} & \ldots & a_{3n} \end{pmatrix}$$

and

$$\mathtt{beq} = \begin{pmatrix} b_3 \end{pmatrix}.$$

The variables should be greater or equal than zero, so

$$\mathtt{lb} = \begin{pmatrix} 0 & 0 & \ldots & 0 \end{pmatrix}.$$

The integer variables are defined by the `intcon` vector and, according to the description, the latter should contain the indexes of the integer variables. For the given example:

$$\mathtt{intcon} = \begin{pmatrix} 1 & 2 & 4 \end{pmatrix}$$

Notice that x_1 is also an integer variable, with the restriction that it must be greater or equal than zero, and lower than or equal to one, that is,

$$x_1 \in \{0, 1\} \Leftrightarrow (0 \leq x_1 \leq 1 \wedge x_1 \text{ integer}) . \tag{6.2}$$

Hence, the vector of the upper bounds is written as

$$\mathsf{ub} = \begin{pmatrix} 1 & \infty & \dots & \infty \end{pmatrix} , \tag{6.3}$$

where ∞ (infinity) is the same as no upper bound.

6.2 DEVELOPED MODEL

After providing this general example, it is now possible to look again at the developed mathematical model and try to fit it to the necessary syntax.

According to the general definitions, the decisions are represented by the variables $x_{l,s}$ (at this point we ignore the specific variables needed for each target), which are set to 1 if load l is allocated to channel s, and set to 0 if otherwise. This suggests a matrix representation of the form:

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N_S} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,N_S} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_L,1} & x_{N_L,2} & \cdots & x_{N_L,N_S} \end{pmatrix} , \tag{6.4}$$

where N_L and N_S are the total number of loads and the total number of channels, respectively. Thus, each row corresponds to a load and each column represents a channel. Constraints (5.1) and (5.2):

$$\sum_{l \in L_s} x_{l,s} \leq 1, \ \forall s \in S \tag{6.5}$$

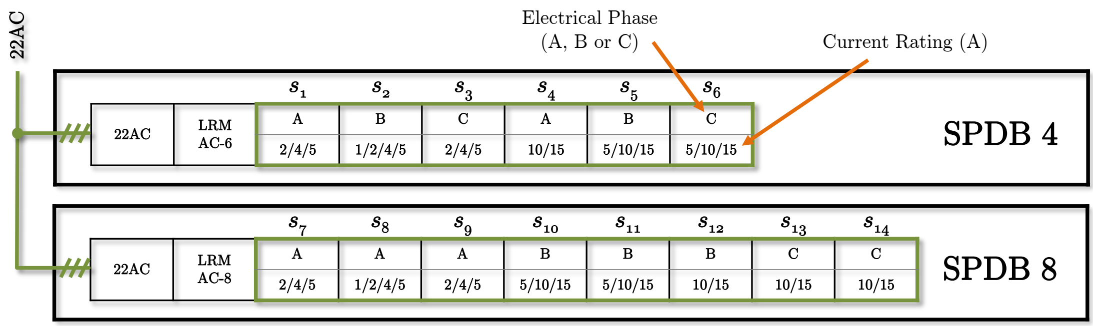
$$\sum_{s \in S_l} x_{l,s} = 1, \ \forall l \in L \tag{6.6}$$

are then quite simple to visualize. The first states that, in each column, no more than one variable can be equal to 1. The second forces that, in each row, exactly one variable is equal to 1. Note that the summations are not performed over the total set of loads and over the total set of channels. This comes from the fact that not every pair load-channel is available. Due to the current rating and electrical phase constraints, some of the variables must be equal to zero. As we will see, this can be achieved using a feasibility matrix:

$$UB = \begin{pmatrix} ub_{1,1} & ub_{1,2} & \cdots & ub_{1,N_S} \\ ub_{2,1} & ub_{2,2} & \cdots & ub_{2,N_S} \\ \vdots & \vdots & \ddots & \vdots \\ ub_{N_L,1} & ub_{N_L,2} & \cdots & ub_{N_L,N_S} \end{pmatrix} , \tag{6.7}$$

where $ub_{l,s}$ is equal to 1, if load l can be allocated to channel s, and set to 0 if otherwise.

Let us consider the following system:



Note that these channels belong to different boxes. Also observe that the channels are numbered in a sequential way. The loads to be allocated have the following specifications:

Load	Optional	AC/DC	Phase	Rating (A)	SPDB (Box)	LRM (Card)	SSPC (Slot)	P_{nom} (W or VA)	$u_{max}(1)$	$u_{op}(1)$	...	Permanent	Sheddable
1	Yes	AC	A	10	4	-	-	300	0.8	0.6	...	Yes	No
2	Yes	AC	B	10	4	-	-	300	0.8	0.6	...	Yes	No
3	Yes	AC	C	10	4	-	-	300	0.8	0.6	...	Yes	No
4	Yes	AC	Any	2	8	-	-	50	0.5	0.5	...	Yes	Yes
4	Yes	AC	Any	2	8	-	-	150	0.2	0.0	...	No	Yes

Observe that the first three loads correspond to a three-phase load. Load 4 is represented by two rows because of the two types of operation mode: permanent and intermittent (see Section 2.2.5). The important parameters for the feasibility matrix are the electrical phase ("Phase"), the current rating ("Rating (A)") and the box ("SPDB (box)") where they must be connected. The following feasibility matrix would then be obtained:

$$
UB = \begin{array}{c} \\ \\ l_1 \\ l_2 \\ l_3 \\ l_4 \end{array}
\begin{array}{c}
\overbrace{\hphantom{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0}}^{\text{SPDB 4}} \quad \overbrace{\hphantom{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0}}^{\text{SPDB 8}} \\
\begin{array}{cccccccccccccc} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} \end{array} \\
\begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

The first three loads can only be connected to SPDB 4. So, for the first three lines of the matrix, the columns corresponding to the channels of SPDB 8 are all set to 0. Similarly, channels inside the SPDB 4 are not available to Load 4, making all columns from SPDB 8 equal to 0.

Regarding the first three loads, they already have a specified electrical phase, since they belong to the same three-phase load (this situation was explained in 2.2.2). This information, together with the necessary current rating, leads to a single allocation possibility for each of them.

Load 4 needs a channel supplying 2 A and no electrical phase is specified. As a result, this load has three allocation possibilities (inside SPDB 8).

The choice of UB to represent the feasibility matrix was not a coincidence. Consider the variable $x_{1,1}$ (allocation of Load 1 to Channel 1). In this example, its value must be 0, since this allocation is not possible. The same is to say that $x_{1,1} = ub_{1,1}$. But variable $x_{4,7}$ (allocation of Load 4 to Channel 7) can be set to 1, since this allocation is possible, but it can also be set to 0 (connecting Load 4 to Channel 7 is not mandatory). Since $x_{4,7}$ is an integer variable, the same is to say that $0 \leq x_{4,7} \leq ub_{4,7}$. Thus, the feasibility matrix defines, in fact, the upper bounds of the variables. Therefore, defining the vector `x` as:

$$\mathtt{x} = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,N_S} & x_{2,1} & x_{2,2} & \dots & x_{N_L,N_S} \end{pmatrix},$$

the vector with the lower bounds `ub` (defined in the previous section) is given by

$$\mathtt{lb} = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix},$$

and the vector `ub` (upper bounds) can be written as

$$\mathtt{ub} = \begin{pmatrix} ub_{1,1} & ub_{1,2} & \dots & ub_{1,N_S} & ub_{2,1} & ub_{2,2} & \dots & ub_{N_L,N_S} \end{pmatrix}.$$

Finally, the input `intcon` must contain the indexes (from vector `x`) of the integer variables:

$$\mathtt{intcon} = \begin{pmatrix} 1 & 2 & \dots & N_L \cdot N_S \end{pmatrix}. \tag{6.8}$$

Constraint (5.1) results in $N_S = 14$ inequalities and is formulated as

$$\begin{pmatrix} \overbrace{\begin{matrix}1&0&0&\dots&0\end{matrix}}^{l=1,\ s=1,\dots,14} & \overbrace{\begin{matrix}1&0&0&\dots&0\end{matrix}}^{l=2,\ s=1,\dots,14} & \dots & \overbrace{\begin{matrix}1&0&0&\dots&0\end{matrix}}^{l=4,\ s=1,\dots,14} \\ \begin{matrix}0&1&0&\dots&0\end{matrix} & \begin{matrix}0&1&0&\dots&0\end{matrix} & \dots & \begin{matrix}0&1&0&\dots&0\end{matrix} \\ \vdots & \vdots & \vdots & \vdots \\ \begin{matrix}0&0&0&\dots&1\end{matrix} & \begin{matrix}0&0&0&\dots&1\end{matrix} & \dots & \begin{matrix}0&0&0&\dots&1\end{matrix} \end{pmatrix} \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,N_S} \\ x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{N_L,N_S} \end{pmatrix} \leq \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Note that according to the description of the MATLAB® function, the first matrix of the above inequality is part of `A`, and the vector of the right-hand side is part of `b`. The same analysis can be performed for constraint (5.2). This constraint leads to $N_L = 4$ equalities:

$$\begin{pmatrix} \overbrace{\begin{matrix}1&1&1&\dots&1\end{matrix}}^{l=1,\ s=1,\dots,14} & \overbrace{\begin{matrix}0&0&0&\dots&0\end{matrix}}^{l=2,\ s=1,\dots,14} & \dots & \overbrace{\begin{matrix}0&0&0&\dots&0\end{matrix}}^{l=4,\ s=1,\dots,14} \\ \begin{matrix}0&0&0&\dots&0\end{matrix} & \begin{matrix}1&1&1&\dots&1\end{matrix} & \dots & \begin{matrix}0&0&0&\dots&0\end{matrix} \\ \vdots & \vdots & \vdots & \vdots \\ \begin{matrix}0&0&0&\dots&0\end{matrix} & \begin{matrix}0&0&0&\dots&0\end{matrix} & \dots & \begin{matrix}1&1&1&\dots&1\end{matrix} \end{pmatrix} \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,N_S} \\ x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{N_L,N_S} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Here, the first matrix is part of `Aeq`, and the vector of the right-hand side is part of `beq`.

Consider that we want to calculate the applicable limit of equation (5.13a). This limit is calculated for the total load consumption on the feeder, so both segments are considered. In terms of the consumption type, this limit involves maximum permanent and maximum intermittent powers. In addition, it considers only non-sheddable loads. This would be formulated as

$$\begin{pmatrix} \overbrace{\begin{matrix}240&0&0&240&0&0&\dots\\0&240&0&0&240&0&\dots\\0&0&240&0&0&240&\dots\end{matrix}}^{l=1,\ s=1,\dots,14} & \begin{matrix}\dots\\\dots\\\dots\end{matrix} & \overbrace{\begin{matrix}0&0&0&0&0&0&\dots\\0&0&0&0&0&0&\dots\\0&0&0&0&0&0&\dots\end{matrix}}^{l=4,\ s=1,\dots,14} \end{pmatrix} \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,N_S} \\ x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{N_L,N_S} \end{pmatrix} \leq$$

$$\begin{pmatrix} 0.87 \cdot m_{22\text{AC},0} - \left(\Omega^{1,\text{m,per},\overline{\text{sh}}}_{1,22\text{AC},0} + \Omega^{1,\text{m,int},\overline{\text{sh}}}_{1,22\text{AC},0}\right) \\ 0.87 \cdot m_{22\text{AC},0} - \left(\Omega^{2,\text{m,per},\overline{\text{sh}}}_{1,22\text{AC},0} + \Omega^{2,\text{m,int},\overline{\text{sh}}}_{1,22\text{AC},0}\right) \\ 0.87 \cdot m_{22\text{AC},0} - \left(\Omega^{3,\text{m,per},\overline{\text{sh}}}_{1,22\text{AC},0} + \Omega^{3,\text{m,int},\overline{\text{sh}}}_{1,22\text{AC},0}\right) \end{pmatrix}$$

There are three inequalities since the constraints must be calculated for each electrical phase. Note that the first line, for example, considers only power consumptions on channels supplying electrical phase A. It is also important to remember that these represent only the first flight phase. For the remaining flight phases, similar inequalities have to be created.

The consumption of Load 1 (and also Loads 2 and 3) is calculated by multiplying the nominal power by the consumption factor ($300 \times 0.8 = 240$, see Equation (2.13)). The table specifies that Load 1 cannot be allocated to channels supplying electrical phase B or C, but still they are included in the inequalities. In this case, the feasibility matrix ensures that these power consumptions will never occur ($ub_{1,2} = ub_{1,5} = \ldots = 0$).

Since Load 4 is a sheddable load, its allocation to some of the considered channels is possible (ones in the feasibility matrix). However, this load does not influence the calculation of this applicable limit. This is the reason why, for this case, all the values must be set to 0.

On the right-hand side, the power consumption of the standard loads is subtracted to the applicable limit. This comes directly from expression (5.13a), but it is also logical, since higher values of standard loads' power consumption lead to a lower margin for optional loads.

Consider now constraint (5.13b). It can be formulated as

$$\begin{pmatrix} \overbrace{\begin{matrix} 240 & 0 & 0 & 240 & 0 & 0 & \dots \\ 0 & 240 & 0 & 0 & 240 & 0 & \dots \\ 0 & 0 & 240 & 0 & 0 & 240 & \dots \end{matrix}}^{l=1,\ s=1,\dots,14} & \begin{matrix} \dots \\ \dots \\ \dots \end{matrix} & \overbrace{\begin{matrix} 55 & 0 & 0 & 55 & 0 & 0 & \dots \\ 0 & 55 & 0 & 0 & 55 & 0 & \dots \\ 0 & 0 & 55 & 0 & 0 & 55 & \dots \end{matrix}}^{l=4,\ s=1,\dots,14} \end{pmatrix} \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,N_S} \\ x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{N_L,N_S} \end{pmatrix} \leq$$

$$\begin{pmatrix} 2 \cdot m_{\mathrm{22AC},0} - \left(\Omega^{1,\mathrm{m,per,sh}}_{1,\mathrm{22AC},0} + \Omega^{1,\mathrm{m,per},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},0} + \Omega^{1,\mathrm{m,int,sh}}_{1,\mathrm{22AC},0} + \Omega^{1,\mathrm{m,int},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},0} \right) \\ 2 \cdot m_{\mathrm{22AC},0} - \left(\Omega^{2,\mathrm{m,per,sh}}_{1,\mathrm{22AC},0} + \Omega^{2,\mathrm{m,per},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},0} + \Omega^{2,\mathrm{m,int,sh}}_{1,\mathrm{22AC},0} + \Omega^{2,\mathrm{m,int},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},0} \right) \\ 2 \cdot m_{\mathrm{22AC},0} - \left(\Omega^{3,\mathrm{m,per,sh}}_{1,\mathrm{22AC},0} + \Omega^{3,\mathrm{m,per},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},0} + \Omega^{3,\mathrm{m,int,sh}}_{1,\mathrm{22AC},0} + \Omega^{3,\mathrm{m,int},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},0} \right) \end{pmatrix}$$

The main difference here, is that the power consumption for Load 4 is now included, both permanent and intermittent ($50 \times 0.5 + 150 \times 0.2 = 55$). This is due to the inclusion of sheddable loads in the calculation of these inequalities.

For the calculation of the applicable limit for the segment supplying SPDB 8 (second segment/box), let us consider that the power limit of this segment is lower than the RCCB rating (case of expression (5.14c)). The formulation is given by

$$\begin{pmatrix} \overbrace{\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{matrix}}^{l=1,\ s=1,\dots,14} & \begin{matrix} \dots \\ \dots \\ \dots \end{matrix} & \overbrace{\begin{matrix} 55 & 0 & 0 & 55 & 0 & 0 & \dots \\ 0 & 55 & 0 & 0 & 55 & 0 & \dots \\ 0 & 0 & 55 & 0 & 0 & 55 & \dots \end{matrix}}^{l=4,\ s=1,\dots,14} \end{pmatrix} \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,N_S} \\ x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{N_L,N_S} \end{pmatrix} \leq$$

$$\begin{pmatrix} 0.87 \cdot m_{\mathrm{22AC},2} - \left(\Omega^{1,\mathrm{m,per},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},2} + \Omega^{1,\mathrm{m,int},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},2} \right) \\ 0.87 \cdot m_{\mathrm{22AC},2} - \left(\Omega^{2,\mathrm{m,per},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},2} + \Omega^{2,\mathrm{m,int},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},2} \right) \\ 0.87 \cdot m_{\mathrm{22AC},2} - \left(\Omega^{3,\mathrm{m,per},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},2} + \Omega^{3,\mathrm{m,int},\overline{\mathrm{sh}}}_{1,\mathrm{22AC},2} \right) \end{pmatrix}$$

Load 4 is the only one included in this calculation, since the remaining have to be allocated in SPDB 4 (first box).

HINTS

- Further variables resulting from the different targets can be added to the `x` vector. The order of the variables is not relevant, but it is very important to check that the variables' coefficients of the inequalities, equalities and objective function (`A`, `Aeq` and `f`, respectively) comply with the order of the variables considered in `x`.

- Repeatedly resizing large arrays often results in decreased performance. This happens, for example, when using `for` or `while` loops. MATLAB® may spend extra time looking for larger contiguous blocks of memory, in order to move the arrays into those blocks [23]. From the author's experience, this has great impact when large matrices are involved. Therefore, specially in the cases of the matrices `A` and `Aeq`, the number of elements should be calculated before creating them, enabling prior allocation of the necessary memory.

- The examples given show that `A` can have a large number of zeros. Although not providing extra information, each of them is still stored, taking up the same amount of memory as any other value. This fact may sometimes lead to high occupation of memory, which deteriorates performance. A possible solution is to use an explicit declaration of sparse matrices [21]. This type of storage includes only the non-zero values and their indexes, making it more efficient in some cases.

CHAPTER 7

Results

In this chapter, some results are presented. First, we will focus our attention on the different expressions developed for some of the constraints. These were related to the three-phase loads and the optional cards. After this, we will look at the general performance of the different optimizations with already fixed formulations.

Before going into their analysis, it is worth mentioning that the results presented here are based on test cases supplied by SILVER ATENA that intend to represent situations similar to the ones where optimization is going to be performed. Also, since it is not possible to show all the results obtained, there is an attempt to exhibit a set of cases that can be regarded as a good sample of the total set of results.

There are some situations where the optimization can take more time than it would be desirable, in a user's point of view. For this purpose, a maximum time of one hour is considered to distinguish successful and unsuccessful optimizations. This was based on an engineering decision. When maximum time is reached, the branch and bound is stopped and the best solution found so far is taken. So, although the optimality of the solution can not be guaranteed, the result can still be used.

Throughout the results, the word "optimization" is used interchangeably to refer to the optimization concept and to a given optimization/allocation problem. The word "allocation" is also used for the latter case.

7.1 Comparison of formulations

7.1.1 Three-phase loads

The constraints affecting three-phase equipment come from the possibility of having connectors that force the three resulting single-phase loads to be connected together, starting on phase A (situation described in section 2.2.2). The possibilities are repeated here:

Formulation P_1

$$\sum_{s_j \in S_{i+1}} j x_{l_{i+1},s_j} - \sum_{s_j \in S_i} j x_{l_i,s_j} \leq 1,\ l_i \in L_T \tag{7.1a}$$

$$\sum_{s_j \in S_{i+1}} j x_{l_{i+1},s_j} - \sum_{s_j \in S_i} j x_{l_i,s_j} \geq -1,\ l_i \in L_T \tag{7.1b}$$

$$\sum_{s_j \in S_{i+2}} j x_{l_{i+2},s_j} - \sum_{s_j \in S_{i+1}} j x_{l_{i+1},s_j} \leq 1,\ l_i \in L_T \tag{7.1c}$$

$$\sum_{s_j \in S_{i+2}} j x_{l_{i+2},s_j} - \sum_{s_j \in S_{i+1}} j x_{l_{i+1},s_j} \geq -1,\ l_i \in L_T. \tag{7.1d}$$

Formulation P_2

$$x_{l_{i+1},s_{j+1}} = x_{l_i,s_j}, \quad l_i \in L_T,\ \forall s_j \in S_i \tag{7.2a}$$

$$x_{l_{i+2},s_{j+2}} = x_{l_i+1,s_j+1}, \quad l_i \in L_T,\ \forall s_j \in S_i. \tag{7.2b}$$

Here, P_1 and P_2 stand for the different formulations that arise when each group of expressions is used together with the remaining constraints. For the explanation of these formulations, please refer to Chapter 5. As defined there, L_T is the set containing the first loads of each group representing a three-phase equipment. The number of three-phase loads for a given optimization problem is given by N_{L_T}. Therefore, it is easy to calculate the total number of inequalities resulting from the set of expressions (7.1):

$$N_{\text{const}} = N_{\text{ineq}} = 4 \cdot N_{L_T}. \tag{7.3}$$

The number of equalities arising from the second set of equations is a bit more difficult to calculate. Here, there are two equalities per three-phase load, multiplied by the number of channels available to each of the loads belonging to set L_T. As described in Chapter 6, the channels available to each load are defined as upper bounds coming from a feasibility matrix. Since the current ratings available on each channel are not possible to predict, the number of equalities is easier to calculate considering that all channels are available. Then, the upper bounds take care of defining the possible connections. As a consequence, the number of equalities is multiplied not by the different N_{S_i} (number of available channels for load l_i), but by $N_S/3$, where N_S is the total number of channels. The division by three comes from the fact that the loads starting the three-phase group can only be allocated to channels supplying phase A. These are channels s_i, with $i = \{1, 4, 7, \ldots N_S - 2\}$. This information is also included in the feasibility matrix, however, it is a fixed requisite that can easily be considered to reduce the number of constraints. The total number of equalities arising from equations (7.2) is then given by:

$$N_{\text{const}} = N_{\text{eq}} = 2 \cdot N_{L_T} \cdot \frac{N_S}{3}. \tag{7.4}$$

Depending on the number of channels, this formulation can contribute to considerably enlarge the set of constraints. As discussed in the previous chapter, large sets of constraints involve large matrices and subsequent high usage of memory, which can contribute to deteriorate the performance of the optimization. Some solutions to this problem were also presented. One of them, the usage of a sparse matrix definition, can be a good way to deal with this case. Sparse matrices are matrices containing a large amount of zeros. In a normal situation, MATLAB® stores these zeros as any other value, using unnecessary memory. Explicitly creating sparse matrices makes their storage much more efficient. Looking again to equations (7.2), we can see that there are only two none zero coefficients per equality. Thus, for each row of the matrix (each constraint), only two of the $N_L \cdot N_S$ values (excluding the variables coming from the distinct targets) are different than zero!

As indicated in Section 3.4, MATLAB® uses the branch and bound algorithm to solve the MIP problems. As explained, the examination of some nodes can be avoided based on the upper and lower bounds found so far. Upper bounds are established by finding feasible points, mainly by heuristic methods, while lower bounds come from LP relaxation, together with some cut generation algorithms. The results of the LP relaxation are tightly connected to the formulation of the problem.

For the given reasons, we focus our attention in the comparison of the number of constraints and on the upper and lower bounds, for each of the formulations. Table 7.1 shows some optimization results using both formulations. It should be stated that, in many situations, the number of the loads is extended by considering that all loads are optional. This enables an increase in complexity, which facilitates the inspection of performance differences. However, this contributes to unsuccessful optimizations (with the considered maximum time). Later, the performance will be evaluated in more detail. For now, this can also be regarded as an opportunity to compare the formulations, by checking which leads to a better solution after a specific interval of time.

Let us first examine Table 7.1, so that we can get an insight into the quantities presented, and also verify some of the differences already calculated. We can start by looking at the values common to both formulations. The first column is only an identifier for the case in study and the second column indicates the target of optimization ("Max" and "Mean" correspond to the first two targets of unbalance optimization given before). As defined in the implementation description, N_L is the number of optional loads and N_S is the number of channels, including those with previously allocated standard loads. The number of feeders is given by N_F and the number of boxes by N_b. The total number of variables is given by N_x. It can be verified that this number equals $N_L \times N_S$, plus the additional variables necessary for each optimization target. For instance, the first case (max optimization) has $N_x = N_L \times N_S + 1$, since variable U from Formulation (5.19) must be added. The feasibility matrix, defined in the previous section, is responsible for selecting the values $x_{l,s}$ that can be different from zero. This includes the verification of the current ratings that each channel can supply, the channels already occupied by standard loads and the electrical phases needed for the three-phase loads. Hence, the value of the true available choices is given by N_{var}. This is calculated by eliminating the variables that have an upper bound equal to the lower bound. Finally, N_{L_T} is the number of three-phase loads.

Now, on to the values that differ, depending on the formulation. N_{ineq} and N_{eq} represent the number of inequalities and the number of equalities for each formulation (P_1 and P_2). This is a good moment to verify the differences in the constraints' number, formerly calculated in Expressions (7.3) and (7.4). Considering the first optimization, the number of inequalities coming from set (7.1), is given by $4 \cdot L_T = 36$, which is exactly the difference between P_1 and P_2 ($90 - 54 = 36$). The number of equalities coming from set (7.2) is given by $2 \cdot L_T \cdot N_S/3 = 216$. This is also equal to the difference between both formulations ($246 - 30 = 216$). Looking at the table, it is clear that formulation P_2 has a total number of constraints ($N_{\text{ineq}} + N_{\text{eq}}$) considerably larger for all cases. The column "Success" identifies the optimizations concluded within the maximum time. "Result" is the obtained value for the optimization and "RelGap (%)" is the relative gap (returned by MATLAB®) and can be regarded as measure of the result's uncertainty. It is calculated by:

$$\text{RelGap}(\%) = \frac{U - L}{|U| + 1} \cdot 100\% \tag{7.5}$$

where U and L are the upper and lower bound of the objective function (respectively), at the end of the optimization. When the best solution found is much greater than the largest lower bound, RelGap $\approx 100\%$, indicating that there is a possibility that a much better solution can be found. When $U - L \to 0$, RelGap $\to 0$, meaning that even if a better solution exists, it can only have a slightly better value. Note that when the optimization is successful (optimal solution found within the established time), the relative gap is equal to zero. The three last quantities presented are: (1) "LP", (2) "Cut" and (3) "Heuristics". The first shows the value of the LP relaxation for node 0 (explained in Section 3.4.2). This is the first lower bound (dual bound). The second shows the lower bound after applying cutting generation algorithms. The third shows a feasible point found on node 0, providing a first upper bound. When cells show "-", it means that the corresponding procedure gave no result.

The results are sorted by optimization target. Within each target, they are divided in two groups, those with significant differences in performance and those with only slight variations. This is represented by different colors in the table. Differences are classified as significant if the result or running time have discrepancies larger than 5%. As it is somewhat evident, if a user is willing to wait for one hour (3600 sec) to get a result, there is no impact when one formulation finishes in 1 sec and the other in 2 sec, even though the difference in performance is clearly over the 5% interval. So this 5% difference will be considered with respect to the maximum time, i.e. the difference in performance is significant if $\Delta t = |t_{P_1} - t_{P_2}|$ is greater than $3600 \times 0.05 = 180$ sec.

Regarding the maximum unbalance optimization (minimization), cases with significant time differences were observed. In all these situations, formulation P_2 concludes the optimization first. There are even some situations (represented by the first case in the table) where an unsuccessful optimization using P_1 is converted into a successful optimization with the usage of P_2. Note that both return the same final result, making the solution from formulation P_1 also an optimal solution. However, if the optimization with P_2 was not performed, we could not assure that this was, in fact, an optimal solution (RelGap = 44.3 %). The optimizations were performed with a non-dedicated computer, which can exhibit varying performance depending on, for example, the running applications. So, time differences must be regarded with some caution. Nevertheless, the number of nodes can be used to support this analysis. Observing the table, the number of nodes examined before reaching the optimal result (or before suspending the branch and bound) is also substantially lower using P_2. There are no differences in the values of the LP relaxation (for node 0) and cutting generation. However, formulation P_2 seems to enhance the search for feasible points. The heuristic methods were able to find solutions in node 0, for all of these cases, when this formulation was used. The same is not verified for P_1. As already explained, finding feasible solutions leads to initial lower upper bounds (primal bounds) that can be used to exclude the examination of some nodes. Moreover, this can also suggest the heuristics are able to find better upper bounds throughout the branch and bound. As it will be discussed in a moment, the difference in performance can also be supported by the improvement in the LP relaxations. Although in these cases there is no observed difference for node 0, this procedure is done in each node, as explained in the branch and bound description. Therefore, if a formulation enables better LP relaxations, this can have a great impact in the performance of branch and bound. The fact that, for this optimization target, as well as for the mean unbalance, the LP is zero almost every time is not a totally unexpected behavior. In the relaxation, variables can take real values. These results only show that it is usually possible to allocate fractions of the loads in such a manner that all of the electrical phases are perfectly balanced.

Considering now the remaining cases for maximum unbalance minimization, even though the differences in performance are below what we previously defined as significant, it is possible to conclude that, for most of the situations, formulation P_2 performs better. This is observed in terms of time (with exception of the optimization with ID 14), and also in terms of explored nodes (with exception of optimizations 8 and 14). These exceptions occur occasionally, with low influence in performance. These results also show that the cutting generation can differ, but no trend is observed. As an example, even though P_1 leads to a stronger lower bound in case 9, the performance is not better (it is even worse). As already verified for the previous cases, formulation P_2 seems to facilitate the search for feasible points, according to the "Heuristics" column.

The last case of this type of optimization shows a situation where both formulations were not able to solve the optimization within the given time. Here, formulation P_1 performed better, since it found a slightly lower unbalance value (approximately 0.2%), enabling a greater level of certainty (lower relative gap). However, this example contains only one three-phase load.

The situations that led to unsuccessful optimizations for both formulations showed no considerable differences in value. Sometimes formulation P_1 performed better, while other times P_2 led to a better solution. For the situations where the optimal solution was found, there was no single case observed where the usage of P_1 led to significant improved performance. On the other hand, many situations were observed (as depicted in the results) where the usage of P_2 significantly enhanced the performance.

When spreading this analysis to the other targets the same behavior is observed. This is important to verify that these conclusions do not depend on the optimization targets.

These results seem to show that formulation P_2 has an overall better performance, making it the best choice for the final implementation. An extra argument supporting this decision can be found by further analyzing these two formulations. The LP relaxation plays an important role in branch and bound. This corresponds to solving the MIP problem, considering that all variables can be real valued. At the end of Section 3.4.2, a definition was presented, in order to compare two formulations:

Given a set $X \subseteq R^n$, and two formulations P_1 and P_2 for X, we say that P_1 is a better formulation than P_2 if $P_1 \subset P_2$.

Let us start by expanding the summations on the left-hand side of inequalities 7.1a and 7.1b, keeping in mind that the feasible matrix priorly defined forces $l_i \in L_T$ to be connected to phase A channels. For the current case, this means that $S_i = \{s_1, s_4, \ldots, s_{N_S-2}\}$. For similar reasons, $S_{i+1} = \{s_2, s_5, \ldots, s_{N_S-1}\}$. Hence,

$$\sum_{s_j \in S_{i+1}} j x_{l_{i+1},s_j} - \sum_{s_j \in S_i} j x_{l_i,s_j} =$$
$$= 2x_{l_{i+1},s_2} + 5x_{l_{i+1},s_5} + \ldots + (N_S - 1)\, x_{l_{i+1},s_{N_S-1}} - x_{l_i,s_1} - 4x_{l_i,s_4} - \ldots - (N_S - 2)\, x_{l_i,s_{N_S-2}}.$$

Assume now that a given solution verifies constraint 7.2a, that is, $x_{l_{i+1},s_{j+1}} = x_{l_i,s_j}$. Substituting in the previous expression:

$$\sum_{s_j \in S_{i+1}} j x_{l_{i+1},s_j} - \sum_{s_j \in S_i} j x_{l_i,s_j} =$$
$$= 2x_{l_i,s_1} + 5x_{l_i,s_4} + \ldots + (|S|-1)\, x_{l_i,s_{N_S-2}} - x_{l_i,s_1} - 4x_{l_i,s_4} - \ldots - x_{l_i,s_{N_S-2}}$$
$$= x_{l_i,s_1} + x_{l_i,s_4} + \ldots + x_{l_i,s_{N_S-2}}.$$

Since the solution must also verify constraint 5.2

$$\sum_{s \in S_l} x_{l,s} = 1,\ \forall l \in L, \tag{7.6}$$

then

$$\sum_{s_j \in S_{i+1}} j x_{l_{i+1},s_j} - \sum_{s_j \in S_i} j x_{l_i,s_j} = x_{l_i,s_1} + x_{l_i,s_4} + \ldots + x_{l_i,s_{N_S-2}} = \sum_{s_j \in S_i} x_{l_i,s_j} = 1.$$

As a result, if $x_{l_{i+1},s_{j+1}} = x_{l_i,s_j}$, constraints 7.1a and 7.1b from formulation P_1 are immediately verified. The same can be deduced for the other two constraints, when $x_{l_{i+2},s_{j+2}} = x_{l_{i+1},s_{j+1}}$. Consequently, since P_1 is the set of feasible solutions (for the LP problem) using the first formulation and P_2 is the set of feasible solutions using the second formulation, we conclude that $P_2 \subseteq P_1$. However, this is not sufficient to assert that P_2 is a better formulation, according to Definition 11 (section 3.4.2). There is still the possibility that $P_2 = P_1$. A possible way to show that this is not true is to find a solution for the LP problem that is feasible using the first formulation and infeasible using second. Table 7.2 shows an example of such a solution.

	A	B	C	A	B	C
	s_1	s_2	s_3	s_4	s_5	s_6
l_1	0.5	0	0	0.5	0	0
l_2	0	0.6	0	0	0.4	0
l_3	0	0	1	0	0	0

Table 7.2: Point that belongs to feasible set P_1 and does not belong to feasible set P_2.

Effectively,

$$\sum_{s_j \in S_{i+1}} j x_{l_{i+1},s_j} - \sum_{s_j \in S_i} j x_{l_i,s_j} = 2 \times 0.6 + 5 \times 0.4 - 1 \times 0.5 - 4 \times 0.5 = 0.7$$
$$\sum_{s_j \in S_{i+2}} j x_{l_{i+2},s_j} - \sum_{s_j \in S_{i+1}} j x_{l_{i+1},s_j} = 1 \times 3 - 1 \times 0.5 - 4 \times 0.5 = 0.5,$$

meaning that this point belongs to feasible set P_1. However, this point does not verify equations (7.2a) and (7.2b). Therefore, $P_2 \subset P_1$ and we conclude that P_2 is a better formulation than P_1.

As verified by the results, a better formulation does not necessarily lead to a better LP relaxation result in all situations. However, it does guarantee that it cannot be worse. The main disadvantage of formulation P_2 is related to the number of constraints. For the given cases (representative of real situations), this did not result in a noticeable deterioration in performance. For all the reasons given above, formulation P_2 was chosen for implementation.

ID	Target	N_L	N_S	N_F	N_b	N_x	N_{var}	N_{L_T}	N_{ineq} P_1	N_{ineq} P_2	N_{eq} P_1	N_{eq} P_2	Success P_1	Success P_2	Result P_1	Result P_2	RelGap (%) P_1	RelGap (%) P_2	t (sec) P_1	t (sec) P_2	N_{nodes} P_1	N_{nodes} P_2	LP P_1	LP P_2	Cut P_1	Cut P_2	Heuristics P_1	Heuristics P_2
1		30	36	1	2	1081	193	9	90	54	30	246	N	Y	297.12	297.12	44.3	0	3600	18.38	636408	4555	0	0	0	-	-	297.12
2		31	48	2	2	1489	212	7	238	210	31	255	Y	Y	190.00	190.00	0	0	545.14	163.36	172062	61355	0	0	0	0	-	190.00
3		29	36	1	2	1045	187	9	90	54	29	245	Y	Y	59.75	59.75	0	0	270.58	56.36	56656	11365	0	0	0	0	-	59.75
4		19	36	2	1	685	169	6	132	108	19	163	Y	Y	488.75	488.75	0	0	187.80	6.02	192247	6620	0	0	0	-	488.75	488.75
5		30	36	1	2	1081	205	9	90	54	30	246	Y	Y	478.15	478.15	0	0	154.27	2.05	34707	497	0	0	0	-	-	478.15
6		28	36	1	2	1009	181	9	90	54	28	244	Y	Y	97.50	97.50	0	0	156.39	6.80	37247	1723	0	0	0	-	-	97.50
7		29	36	1	2	1045	187	9	90	54	29	245	Y	Y	97.50	97.50	0	0	56.36	3.67	14994	777	0	0	0	0	97.50	97.50
8	Max	28	48	2	2	1345	176	7	220	192	28	252	Y	Y	299.75	299.75	0	0	103.08	72.64	27714	31307	0	0	0	314.75	-	0
9		22	48	2	2	1057	176	7	148	120	22	246	Y	Y	488.75	488.75	0	0	69.97	53.61	56658	54828	0	0	202.08	54.31	488.75	488.75
10		25	36	1	2	901	151	8	86	54	25	217	Y	Y	44.82	44.82	0	0	1.14	0.75	329	8	0	0	0	0	-	44.82
11		25	36	1	2	901	151	8	86	54	25	217	Y	Y	0	0	0	0	0.47	0	65	0	0	0	0	0	-	0
12		24	36	1	2	865	145	8	86	54	24	216	Y	Y	0	0	0	0	0.30	0	40	0	0	0	0	-	-	-
13		30	36	1	2	1081	193	9	90	54	30	246	Y	Y	162.06	162.06	0	0	2.04	1.12	672	199	0	0	0	-	-	162.06
14		29	36	1	2	1045	187	9	90	54	29	245	Y	Y	305.62	305.62	0	0	4.00	7.97	1095	1968	0	0	11.75	-	305.62	305.62
15		6	57	2	2	343	17	2	174	166	6	82	Y	Y	490.94	490.94	0	0	0	0	0	0	489.85	490.94	-	-	-	-
16		26	36	1	2	937	389	1	115	111	26	50	N	N	228.20	228.60	0.57	0.74	3600	3600	313827	377014	0	0	0	-	-	392.17
17		25	48	2	2	1232	231	8	296	264	25	281	Y	Y	106.91	106.91	0	0	610.16	193.88	272489	94984	0	0	0	0	-	106.91
18		31	48	2	2	1520	225	8	344	312	31	287	Y	Y	101.98	101.98	0	0	279.44	90.78	74790	28168	0	0	0	0	233.92	220.83
19		29	36	1	2	1060	201	9	168	132	29	245	Y	Y	239.62	239.62	0	0	108.47	4.22	23836	880	0	0	0	-	-	239.62
20	Mean	29	36	1	2	1060	201	9	168	132	29	245	Y	Y	116.02	116.02	0	0	17.06	7.17	4544	1552	0	0	0	0	116.02	116.02
21		25	36	1	2	916	165	8	164	132	25	217	Y	Y	42.02	42.02	0	0	8.05	0.97	2582	177	0	0	0	-	42.02	42.02
22		24	36	1	2	880	159	8	164	132	24	216	Y	Y	0	0	0	0	0.41	0	30	0	0	0	0	-	-	-
23		35	45	1	2	1591	521	2	209	201	35	95	N	N	169.88	173.17	4.22	52.72	3600	3600	279961	279349	0	0	0	-	293.70	385.87
24		38	60	1	2	2296	656	4	226	210	38	198	N	N	16.78	16.61	79.00	39.10	3600	3600	149717	137305	0	0	0		-	501.95
25		25	48	2	2	1228	227	8	184	152	31	287	Y	Y	160	160	0	0	417.25	7.85	298780	8641	73.82	73.82	73.82	73.82	-	-
26		25	36	1	2	914	164	8	113	81	28	220	Y	Y	170	170	0	0	96.15	0.64	79233	12	64.95	64.95	94.28	90.55	-	-
27		35	45	1	2	1589	519	2	261	253	38	98	Y	Y	210	210	0	0	57.76	5.53	8	76	63.66	63.66	152.62	90.00	-	-
28	Weight	29	36	1	2	1058	200	9	117	81	32	248	Y	Y	330	330	0	0	17.77	5.99	1551	3272	94.92	94.92	235.01	195.01	-	-
29		38	60	1	2	2294	654	4	266	250	41	201	Y	Y	170	170	0	0	3.66	11.63	143	9	74.10	74.10	90.00	150.27	-	-
30		29	36	1	2	1058	200	9	117	81	32	248	Y	Y	370	370	0	0	3.37	0.88	37	18	134.71	134.71	198.50	183.74	-	-
31		29	36	1	2	1058	200	9	113	81	27	219	Y	Y	250	250	0	0	0.69	0.48	25	20	97.64	97.64	99.58	98.89	-	-
32		31	48	2	2	1504	209	8	200	168	31	287	Y	Y	103	103	0	0	37.20	0	678	0	99.00	99.00	102.00	103.00	103.00	104.00
33	Alloc Poss	35	45	1	2	1590	520	2	158	150	35	95	Y	Y	106	106	0	0	7.05	8.31	65	9	105.67	105.67	105.67	105.67	-	108.00
34		25	48	2	2	1216	215	8	152	120	25	281	Y	Y	88	88	0	0	3.71	0	4917	0	86.00	86.00	86.00	88.00	-	88.00
35		29	36	1	2	1056	198	9	108	72	29	245	Y	Y	53	53	0	0	0.42	0.09	211	10	52.00	52.00	52.00	52.00	-	-

Table 7.1: Comparison of results obtained with the different formulations for the three-phase loads' allocation constraints.

7.1.2 OPTIONAL CARDS

The same type of analysis was carried out for the different formulations responsible for modeling the optional cards' constraints. Two of them were related:

Formulation P_1

$$\sum_{l\in L_s} x_{l,s} \leq 1 - \sum_{l\in L_\nu} x_{l,\nu}, \quad i = 1,\ldots,N_{\mathrm{opt}_k},\ \forall s \in S_k^{\mathrm{opt}_i},\ \forall \nu \in S_k^{\mathrm{opt}} \setminus S_k^{\mathrm{opt}_i} \tag{7.7}$$

Formulation P_2

$$\sum_{s\in S_k^{\mathrm{opt}_i}} \sum_{l\in L_s} x_{l,s} \leq N_{S_k^{\mathrm{opt}_i}} \left(1 - \sum_{l\in L_\nu} x_{l,\nu}\right), \quad i = 1,\ldots,N_{\mathrm{opt}_k},\ \forall \nu \in S_k^{\mathrm{opt}} \setminus S_k^{\mathrm{opt}_i} \tag{7.8}$$

The third included extra variables:

Formulation P_3

$$\sum_{s\in S_k^{opt_i}} \sum_{l\in L_s} x_{l,s} \leq N_{S_k^{opt_i}} r_{i,k}, \quad i = 1,\ldots,N_{\mathrm{opt}_k} \tag{7.9a}$$

$$\sum_{i=1}^{N_{\mathrm{opt}_k}} r_{i,k} \leq 1, \quad k = 1,\ldots,N_K \tag{7.9b}$$

$$r_{i,k} \in \{0,1\} \tag{7.9c}$$

Formulation P_2 is obtained by combining inequalities from Formulation P_1. Consider now a single optional card (N_{opt_k}) and only two card options (with consecutive numbered channels). Using the following notation:

$$N_{S_k^{\mathrm{opt}_i}} := N_i$$

$$\sum_{l\in L_\mu} x_{l,\mu} := A_\mu(x), \quad \mu = 1,\ldots,N_1$$

$$1 - \sum_{l\in L_\nu} x_{l,\nu} := 1 - B_\nu(x), \quad \nu = 1,\ldots,N_2$$

the second formulation for this case can be rewritten as

$$\sum_{\mu=1}^{N_1} A_\mu(x) \leq N_1 \left(1 - B_\nu(x)\right), \quad \nu = 1,\ldots,N_2.$$

For a given value of ν, the inequality can be presented as

$$A_1(x) \leq N_1 \left(1 - B_\nu(x)\right) - \sum_{\mu=2}^{N_1} A_\mu(x).$$

From the first formulation:

$$A_\mu(x) \leq 1 - B_\nu(x), \quad \mu = 1,\ldots,N_1,$$

thus,

$$N_1 \left(1 - B_\nu(x)\right) - \sum_{\mu=2}^{N_1} A_\mu(x) \geq 1 - B_\nu(x).$$

This means that:

$$A_1(x) \leq 1 - B_\nu(x) \leq N_1 \left(1 - B_\nu(x)\right) - \sum_{\mu=2}^{N_1} A_\mu(x). \tag{7.10}$$

This result can be extended to the general case, proving that $P_1 \subseteq P_2$. A simple example can be found for a point contained only in P_2, making P_1 a better formulation. Regarding the third formulation, the analysis is not straightforward, since extra variables are considered.

The results obtained show only minor differences, and none of the formulations achieved clearly better results. There are examples with slightly faster performance for each of the different formulations. Since P_3 helps implementing the objective functions of weight optimization, this was the chosen formulation to be included in the final tool.

7.2 OPTIMIZATION RESULTS

The results for the various optimizations are presented in this section. For this purpose, 12 sets of test data are analyzed. These correspond to manual allocations performed by SILVER ATENA and each of them involves various feeders and boxes, AC and DC. Some of the feeders comprise distinct groups of loads, and as a result they can be optimized separately. Thus, for each of these 12 sets, several optimizations are performed. The results are going to focus unbalance and weight optimization. The optimization of further allocation possibilities was also tested and implemented in the developed tool. However, its usage will be normally considered as the last objective function of a preemptive optimization.

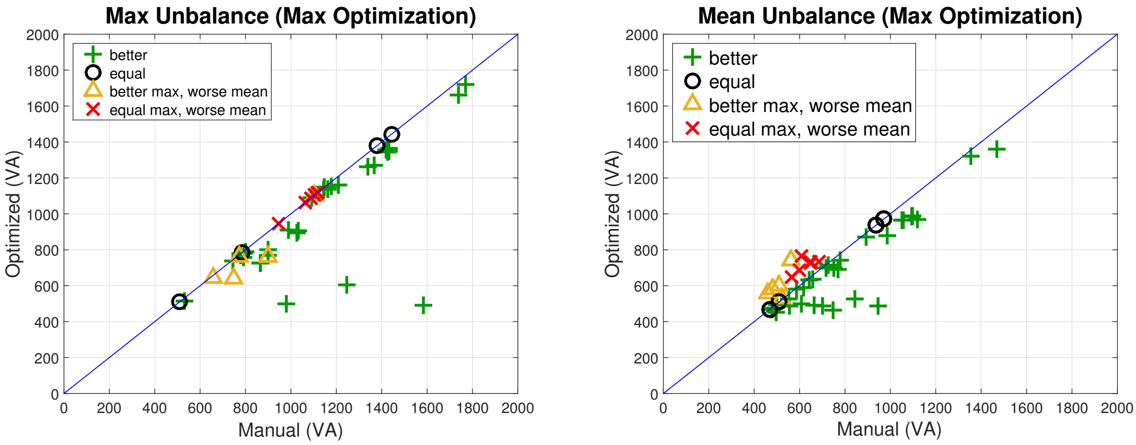
During the results, some plots have "Optimization ID" as the label for the x-axis. This means optimization identification, and it serves merely to show the different optimization problems. Therefore, their order is chosen based on the criterion in study and changes during the results' presentation.

7.2.1 FIXED SYSTEM

The first results are obtained considering the system is exactly the same as the one used for the manual customizations. This means all the cards and cables are considered fixed. The decisions are confined to the choice of channel for each load. For this reason, we focus our attention on the results obtained for the three-phase power unbalance (weight optimization is not a possibility). The defined targets were (see section 4.1.2):

1. Minimize the maximum unbalance among all feeders and all flight phases. (Max Optimization)
2. Minimize the mean unbalance among all feeders and all flight phases with the possibility of assigning weights to the flight phases. (Mean Optimization)
3. Minimize the maximum unbalance among all feeders, considering the powers on each electrical phase the maximum among all flight phases.

Only the results of the first two optimization targets are analyzed. As previously mentioned, the third target was mainly used for verifying the developed mathematical model and is not particularly interesting in terms of real application. The second target will be used throughout the results, with equal weights for every flight phase (no preference is given for specific flight phases). Also note that these targets are not applicable for DC optimization.



(a) (b)

Figure 7.1: Minimization of the maximum unbalance among all feeders and all flight phases. Part (a) shows the maximum unbalance before and after optimization, and (b) depicts the mean unbalance obtained for the same allocations.

Figure 7.1 depicts the results obtained for 46 optimizations with the first target referred above. Each point represents an optimization/allocation problem, where the x-value is the result from the manual procedure ("Manual (VA)") and the y-value is the result after optimization ("Optimized (VA)"). They mainly address situations with a single feeder and multiple boxes. Figure 7.1a shows the comparison results for the worst case of unbalance among all feeders and all flight phases. This corresponds exactly to the result of the objective function. Since the target is to minimize the unbalance value, points below the line correspond to improvements in the objective value. Points on the line represent situations where the unbalance value is exactly the same before and after optimization. Finally, points above the line show situations where the result obtained is worse when compared to the previous manual allocation.

Figure 7.1b displays the value of the mean unbalance among all feeders and all flight phases for the same allocations. The same line is depicted, in order to compare the results before and after optimization. Note, however, that the optimization was not performed according to this target.

The first thing to observe when examining the results in Figure 7.1a is that no point is located above the line. This is, in fact, the expected result. If an optimization was completed and the result was worse than the previous allocation scheme, this would mean that the program was not working correctly. There would exist at least one better solution than the (non-optimal) one obtained. On the other hand, the mean value can be worse than the previous allocation (see Figure 7.1b). This is exactly because the optimization was not performed for this target. It is also possible to see in these figures that the values of the max and mean unbalance are not equally distributed in value. The mean unbalance is mainly located in the interval 400 to 800 VA, while the max unbalance is spread over the interval 400 to 1800 VA. Since these optimizations mainly involve a single feeder, this is an evidence that the power consumptions of the loads can vary strongly with respect to the flight phases. If this was not the case, the values of max and mean unbalance would be similar. Moreover, this can also indicate that optimizing the mean unbalance can lead to different allocations from the ones obtained.

Since the result of the maximum unbalance is always equal or better (when compared to the manual allocation), four types of situations were identified, taking into account the values of the mean unbalance:

- *better*: situations where both values (max and mean) are lower than before, or the cases where one of the values is lower and the other remains the same.
- *equal*: cases where both values (max and mean) are equal before and after optimization.

- *better max, worse mean*: as the name indicates, these are situations where the max value is lower, but the mean value is larger than before.
- *equal max, worse mean*: situations that result in a worse allocation than before, since the max value is the same, but the mean value is larger.

The first group can be regarded as a clear enhancement to the previous allocation, since it improves both values. Most of the optimizations lie in this group: 31/46. This does not mean that there is no better solution for the mean unbalance. There can be allocations with the same max unbalance (never lower), but better mean unbalance.

The second group is not an enhancement. However, there can be situations where the manual allocation was in fact the best possible solution for both these values. But at the moment this cannot be guaranteed, for the same reason presented above: this is the best solution in terms of max unbalance, but the mean value can eventually be improved. There were 4/46 allocations of this type.

The third group (5/46) presents allocations that improved the max unbalance, but led to deterioration of the mean unbalance. This can also be considered as an improvement with respect to the max unbalance.

The last case (6/46) is the most problematic, since there is at least one better allocation. These situations still have the best value for max unbalance, which means that no problem in the optimization occurred. Nevertheless, there must be more solutions with the same value of max unbalance. One of them is the one that was previously obtained with the manual allocation.

The results obtained show that multi-objective optimization should be considered in order to improve some of the allocations. This will be discussed later.

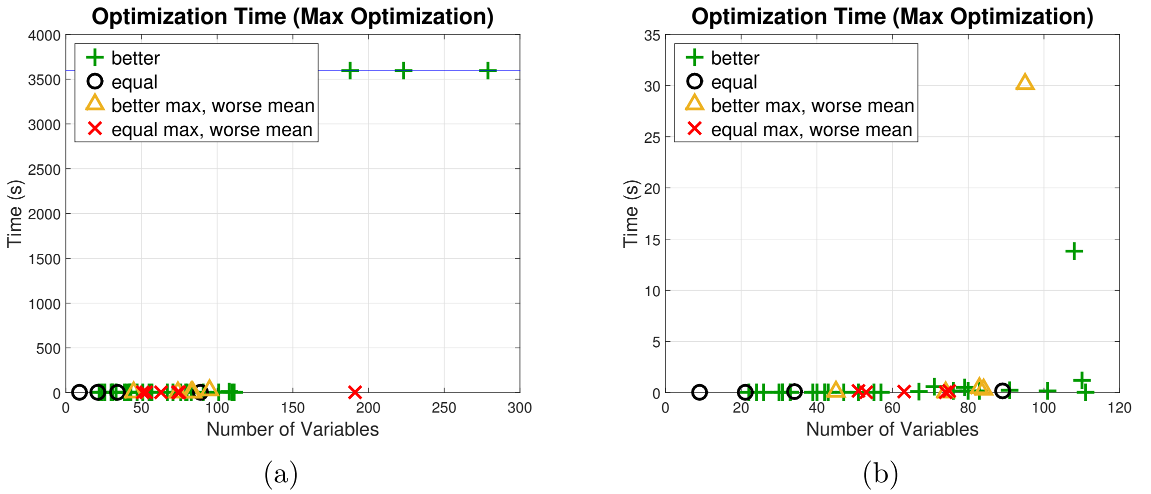


Figure 7.2: Minimization of the maximum unbalance among all feeders and all flight phases. Part (a) shows the time for completion versus the number of variables, for each optimization, and (b) represents the same data, focusing on the optimizations with lower number of variables.

For the moment, let us look at the optimization time for these allocations. Figure 7.2a shows the time that the program took to finish the optimization, for each allocation problem, versus the number of variables of the problem. Once again, only variables with upper bound greater than the lower bound are considered. The line in Figure 7.2a indicates the maximum allowed time (1 hour). Figure 7.2b depicts the same data, but focusing on the optimizations with lower number of variables. These two graphics show a general increase in the duration of the optimization when the set of variables enlarges (with some exceptions). This is a somewhat expected result. When branch and bound is applied (see Section 3.4.2), the number of variables can increase considerably the amount of nodes to explore, boosting the optimization's duration. The growth in complexity is not linear. Thinking in terms of binary decisions, each additional variable doubles the number of possibilities. This is not exactly true for the present case, since there are other constraints that make this calculation not so simple. However, as it can be observed in the figures, the increase in time is clearly non-linear. Until approximately 100 variables, the optimization is practically immediate (in a user's point of view). When the problem has around 100 variables, the time starts increasing. When this value is raised further, unfinished optimizations start to appear.

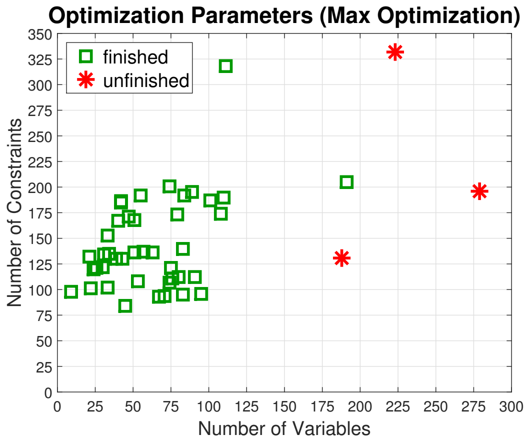


Figure 7.3: Minimization of the maximum unbalance among all feeders and all flight phases. Number of constraints against the number of variables.

In Figure 7.2a, there is one particular case, with approximately 200 variables, that does not follow this tendency. It is interesting to observe that it has approximately the same number of variables as one of the unfinished optimizations. The number of constraints in each of these optimizations can be important to give some more clues on this particular case. Figure 7.3 depicts the number of variables and constraints for each of the allocations. Cases located in the lower-right corner are potentially the worst scenarios for optimization. They have a larger number of variables and lower number of constraints. Constraints may help eliminating possibilities, and consequently diminishing the number of feasible solutions. This can help explaining the difference in the two cases mentioned. The unfinished optimization has approximately 125 constraints, while the completed optimization has a few more than 200 constraints. This difference is related to three-phase loads. The number of constraints does not explain every situation by itself. Distinct constraints eliminate different numbers of allocation possibilities, so it is erratic to make assumptions considering only the number of variables and constraints. As described in Section 3.4.2, we can even be lucky in that the LP relaxation on the root node immediately finds the optimal solution for the MIP problem. This can even happen for a problem with ten times more variables and ten times fewer constraints than the worst case depicted here. Nonetheless, there are many situations where this does not happen, and so the number of variables and constraints is certainly an aspect to keep in mind. As we will discuss later, there are some situations where a lower number of feasible points can also have negative impact in the branch and bound.

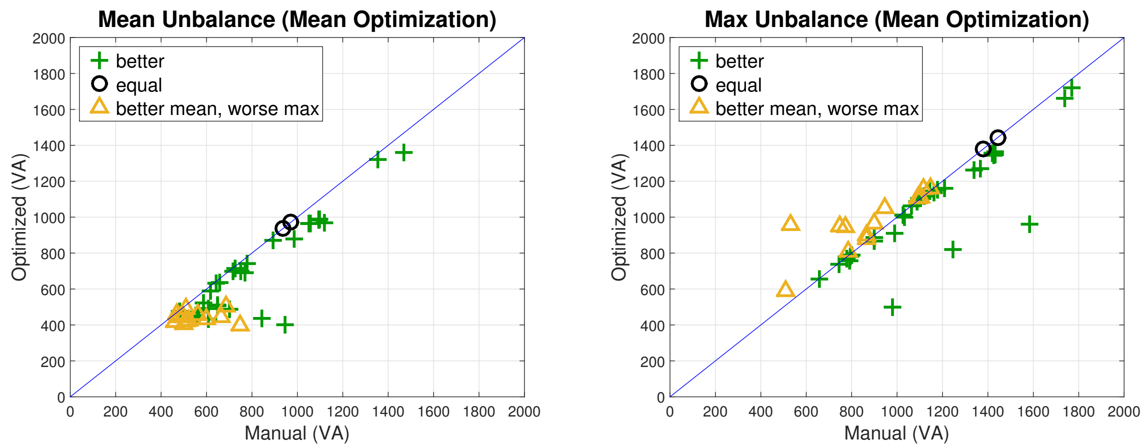
It would be also appealing to find a relation between the running time of the optimization versus the number of loads, or number of channels. It would be interesting in a user's point of view, to somehow predict the necessary time for a given optimization. However, as previously stated, this system has constraints related to current ratings and other factors that can substantially decrease the number of feasible solutions. This restrains the possibility of finding correlations between these quantities. On the other hand, the number of variables already considers most of these constraints, with the upper bounds from the feasibility matrix described in the previous chapter.

Looking again at Figure 7.2a, there were 3/46 unfinished optimizations. However, these cases coincide with improved allocations, compared to the previous manual ones. More interesting is the fact that these three situations correspond to the three strong improvements depicted in Figure 7.1a (points far from the line). This is not a coincidence. The fact that optimality is not guaranteed, can be linked to the increased complexity of the problem, revealed in the number of variables. This increased complexity also makes the manual allocation more difficult, leading to possible worse results in terms of phase balancing. Thus, there is more margin for improvement in these cases.

It should be mentioned that the relative gap for all these unfinished cases is below 2%, which indicates that even if there is a better value, the difference is not larger than 2%. Moreover, the final value was found within the first 30 seconds of the branch and bound. Consequently, there was no benefit in running the optimization for 1 hour. The same result would be obtained if the maximum time was set to 1 minute (for example). This can be a sign that the optimizations could be performed with a lower time limit.

MEAN OPTIMIZATION

The same type of analysis can be done for the mean unbalance optimization (second target). Figure 7.4 shows the obtained results for the mean and max unbalance.

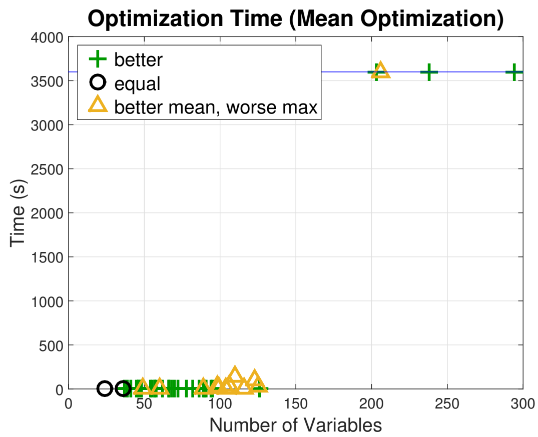


(a) (b)

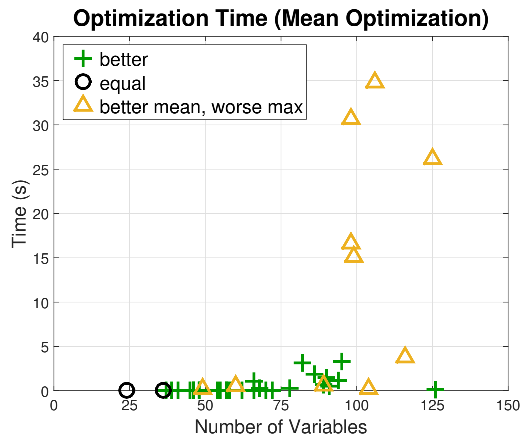
Figure 7.4: Minimization of the mean unbalance among all feeders and all flight phases. Part (a) shows the mean unbalance before and after optimization, and (b) depicts the maximum unbalance obtained for the same allocations.

The results are grouped in the same manner as before, with the respective changes according to the new optimization target: *better*, *equal*, *better mean, worse max*, *equal mean, worse max*. These comparisons are made again regarding the original manual allocations.

As described before, the last group leads to a clearly worse allocation than before (in terms of unbalance), since one of the values is equal and the other is larger. Using this optimization target, there was no allocation with this outcome. This optimization was able to improve 31/46 allocations and 2/46 remained exactly the same. However, there are 13/46 situations where the max unbalance is larger than before.



(a)



(b)

Figure 7.5: Minimization of the mean unbalance among all feeders and all flight phases. Part (a) shows the time for completion versus the number of variables, for each optimization. Part (b) represents the same data, with focus on the optimizations with lower number of variables.

The running times of the different optimizations are depicted in Figure 7.5. The behavior is similar to the max optimization case. Until approximately 100 variables, the optimization is practically instantaneous. After this value, times start to increase to a point where the optimizations are not completed. This is also visible in Figure 7.6.

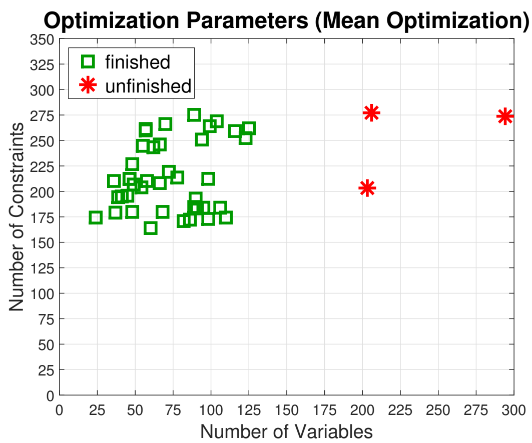


Figure 7.6: Minimization of the mean unbalance among all feeders and all flight phases. Number of constraints against the number of variables.

The incomplete optimizations also reached the final result long before the time limit. The worst case was around 200 seconds.

As already observed, 13/46 results led to an improvement when compared with the original manual allocation. Yet, this result does not take into account the previous optimizations regarding maximum unbalance. It is important to know that we are able to improve a manual allocation. However, it is also relevant to compare the values with other results that we already proved to be achievable. Figure 7.7 compares the maximum unbalance obtained with both optimizations.

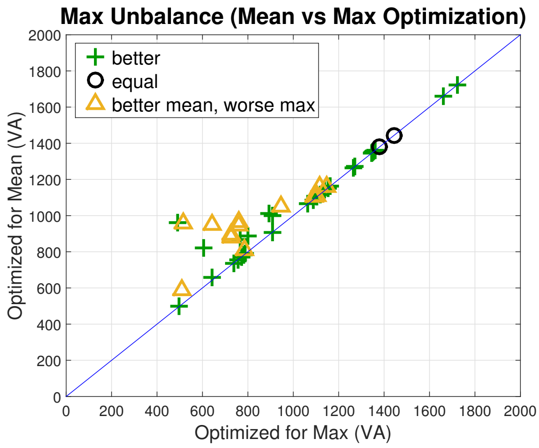


Figure 7.7: Maximum unbalance. Comparison of the results obtained using max and mean optimization targets.

Comparing this plot with the one from Figure 7.4b, it is noticeable that the mean unbalance optimization is not exploring the whole capability of the program to improve the worst case of unbalance. This is a natural consequence of optimizing with respect to only one of the targets and, even if a further max optimization is done, there will probably exist some cases where the maximum unbalance will be still higher than the best possible value. This means that, in some situations, it is not possible to achieve the overall best max unbalance with the desired value of mean unbalance.

A similar conclusion could be performed by comparing the mean unbalance of both types of optimization. But there is a good reason to focus on the maximum unbalance. In Section 4.1.2, it was stated that priority is given to reducing high values of unbalance. This means that the main objective should be to improve the worst cases, which is exactly what is done when max optimization is performed. Nevertheless, there can be situations where a further mean optimization, keeping the max unbalance fixed, lead to a better overall result.

PREEMPTIVE OPTIMIZATION

For the reasons described above, preemptive optimization is done considering the following priority:

1. Minimize the maximum unbalance among all feeders and all flight phases. (Max Optimization)
2. Minimize the mean unbalance among all feeders and all flight phases with the possibility of assigning weights to the flight phases. (Mean Optimization)

The results obtained are presented in Figure 7.8.

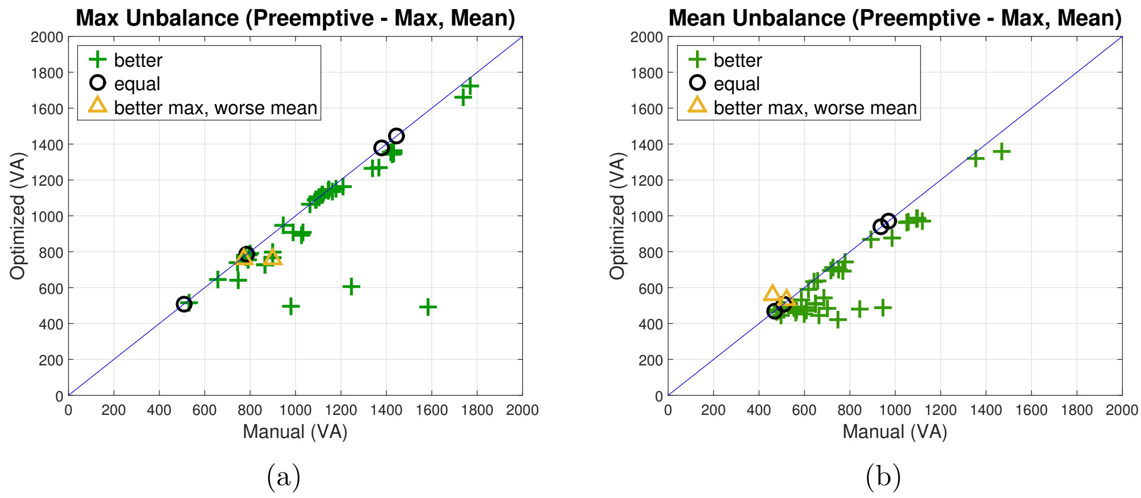


Figure 7.8: Preemptive Optimization: Max Optimization followed by Mean Optimization. Part (a) shows the maximum unbalance before and after optimization, and (b) depicts the mean unbalance obtained for the same allocations.

The results are compared with the original manual allocation. Examining Figures 7.8 and 7.1, several allocations are improved. Instead of 31/46, there are now 40/46 better allocations. The number of allocations with the same values for unbalance remains the same (4/46). There are now no optimizations resulting in worse overall allocations (before there were 6/46). Finally, the number of allocations with better max but worse mean was reduced to 2/46. Note that the location of the points in Figures 7.8a and 7.1a remain exactly the same, since the maximum unbalance is the same for both types of optimization. The difference lies in the type of points, due to the improvement of the mean unbalance. This can be verified by inspecting Figures 7.8b and 7.1b.

The fact that no optimization results in worse overall allocations is a consequence of the preemptive optimization. As referred in Section 3.5.1, the outcome of this method is an efficient point, i.e. there is no result with better performance in one of the targets without worsening the other. Moreover, since optimization was done first with respect to maximum unbalance, the best value for max is ensured. Thus, a better value of mean unbalance cannot be found without worsening the value of max. As a result, no optimization can be part of the "equal max, worse mean" group.

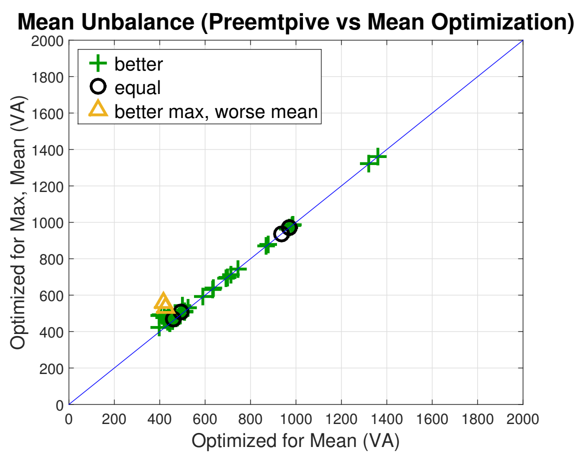


Figure 7.9: Mean unbalance of each allocation problem. Comparison of the results obtained using preemptive optimization (max, mean) and mean optimization.

A comparison can also be made between the results for the mean unbalance (with multi-objective), and the ones previously obtained, when single-objective mean optimization was performed. These are plotted in Figure 7.9. The results show that most of the allocations attain a mean unbalance close to the overall optimal value.

The results observed so far lead to some possible conclusions: (1) Optimizing with respect to max or mean leads to different allocations, and distinct values for these quantities. (2) Performing preemptive optimization can contribute to substantial improvements. This indicates that there is usually more than one allocation with the same value of maximum unbalance (same applies for mean unbalance).

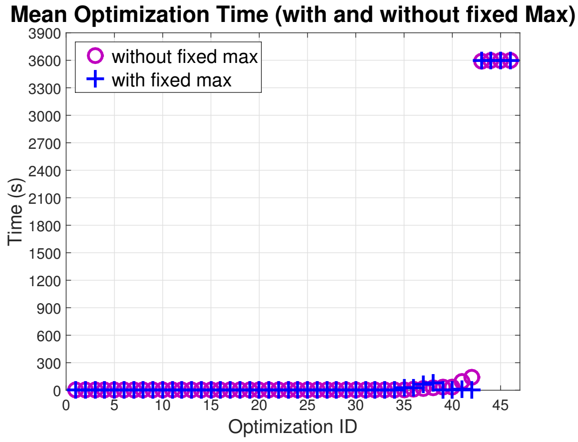


Figure 7.10: Running time for mean optimization with and without prior max optimization.

Regarding time, an analysis can be made by comparing the times of the mean optimizations, with and without prior max optimization. This is done in Figure 7.10. The optimizations are ordered by time of the single-objective optimizations (without fixed max). The duration of the optimizations is approximately the same for both cases (with and without fixed max). A decrease in optimization time could be expected in the case with fixed max. As described in Section 3.5.1, the fixed max is obtained by adding an extra constraint to the mean optimization. This eventually reduces the number of feasible solutions of the problem. There are some situations where this is observed, but there are also some cases where the optimization with this extra constraint takes in fact more time than before. Special attention must be devoted to the cases where the time limit is reached. Some of them also reach the time limit when max optimization is done. Hence, in the case of preemptive optimization, this will lead to situations with a total time of 2 hours. A possibility would be to restrict each optimization to a maximum of 30 minutes. Another interesting possibility involves the weighted sum of objectives (described in Section 3.5.2).

It should be clear at this point that multi-objective optimization must be performed. Considering an aircraft is going to operate all its lifetime with this allocation scheme, the possible increase in optimization time should be largely compensated by the enhancement of the allocation parameters. Even if time limits are an issue, several situations can be improved using lower running times.

WEIGHTED SUM OF OBJECTIVES

The optimizations of maximum and mean unbalance involve the same type of magnitude with the same units. Therefore, the weighted sum of objectives is particularly straightforward. Consider the following minimization problem:

$$\text{minimize} \quad w_{\text{max}} u_{\text{max}}(x) + w_{\text{mean}} u_{\text{mean}}(x) \tag{7.11}$$

where $u_{\text{max}}(x)$ and $u_{\text{mean}}(x)$ are the objective functions of max and mean optimization, respectively. The importance of each of them is represented by the constants w_{max} and w_{mean}.

This weighted sum optimization can be used to approximate a preemptive optimization. This is done by setting w_{max} much greater than w_{mean}. These values should be chosen in order to always give preference to the improvement of $u_{\text{max}}(x)$. At first glance, there is no apparent problem in choosing w_{mean} as large as possible (comparing to w_{mean}). However, since these optimizations are performed using a computer, there are always rounding and/or truncation operations (there are no computers with infinite memory). So, it is usual for the optimization to be considered as finished when the difference between the best solution found (upper bound) and the lower bound is smaller than a chosen value. This was presented in Equation (3.17).

The function `intlinprog` offers different possible stopping criteria. The standard option is based on Equation (7.5). The optimization stops when RelGap is lower than $1 \cdot 10^{-4}$ (or the time limit is reached). If we set w_{max} too large, it can happen that possible improvements in the mean unbalance are below this relative gap. Hence, even though the best possible max unbalance is found, there would be better solutions in terms of mean unbalance (with the same max). However, the optimization does not have enough precision to detect them. The value for the stopping criterion could be reduced, making the solution more accurate, but this poses other types of problems. Optimizations can then become extremely time-consuming.

After testing different possibilities, the following values were chosen: $w_{\text{max}} = 100$ and $w_{\text{mean}} = 1$ (the stopping criterion was not changed). This is equivalent to stating that an improvement of 1 VA in terms of maximum unbalance has the same impact on the final result as reducing 100 VA the mean unbalance. These values lead to weighted sum optimizations with the same results as the preemptive optimization.

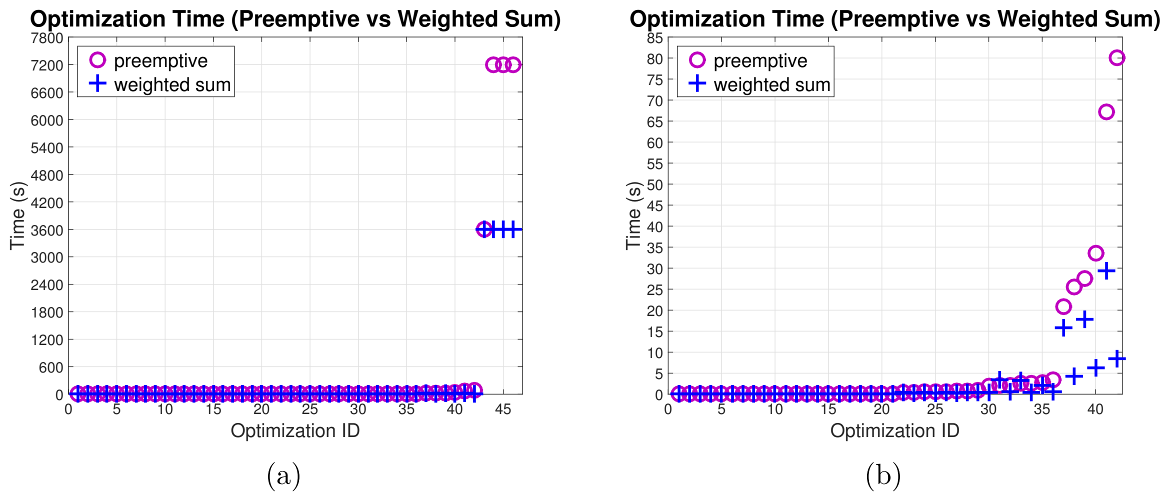


Figure 7.11: Running time for multi-objective optimization: preemptive versus weighted sum objectives (a) shows the time for completion for all allocation problems. (b) represents the same data, focusing on the optimizations with lower number of variables.

In Figure 7.11, the times obtained with this method are compared with the total times from preemptive optimization (max optimization plus mean optimization). Once again, the figure on the right hand side presents the same data, but focusing on a specific region.

The weighted sum method has a slightly better overall performance. The main advantage comes exactly from the cases where the time limit was reached. For preemptive optimization the total time is now 2 hours, while for the weighted sum the maximum time is kept at 1 hour. It would be interesting to try comparing these two possibilities with optimizations completed slightly below the time limit. This could be important to verify if there is effectively an advantage in performing weighted sum optimization with a 1 hour time limit, when compared to preemptive optimization with a 30 minute time limit for each optimization. Since the final results were reached long before these limits, this cannot be verified.

Due to its increased precision, the preemptive optimization was the selected method to incorporate the final tool. However, the weighted sum method enables more versatility (different importances can be assigned to the targets). But there are some situations where the weighted sum leads to a slightly worse result than the preemptive optimization (below $1 \cdot 10^{-4}\%$). The best tuning for the weights of the two targets depends on the allocations to optimize.

For the rest of the results, when unbalance optimization is mentioned it means preemptive method with first max and then mean unbalance optimizations.

SUMMARY

For the optimization of the unbalance in a fixed system, the preemptive method was used. First maximum unbalance and then mean unbalance are optimized. The final unbalance optimization led to average reductions of about 8% in the maximum unbalance and 11% in the mean unbalance, when compared to the manual allocations. In some situations, the reductions can go up to approximately 70% and 50%, respectively.

Most of the optimizations are completed within two minutes. Some of them are practically immediate (in a user's point of view). A minority (4/46) of the optimizations reach the time limitations, for one or both targets. However, the best result is obtained within the first few minutes of optimization. Hence, if time limitations are an issue, the running times can eventually be reduced, without affecting the overall outcome.

7.2.2 OPTIONAL CARDS

Before presenting the results, it is worth taking some time to understand the implications of optional cards. When the optimization is performed, some of the loads are standard and some are optional. As already stated, standard loads have a predefined allocation. Therefore, the cards where they are allocated must be kept fixed. However, if a card is only used to connect optional loads, we consider that it can eventually be changed.

The choice of the (LRM) card is also made by the optimization program (as described in Chapter 5), in order to obtain the best possible result (with respect to the chosen objective function). This is a step higher in the customization process (see Figure 1.2).

When unbalance optimization is performed, the results can never be worse than the same optimization, keeping the cards fixed. This is immediately verified by the fact that the latter is a possible solution for the optimization with optional cards.

UNBALANCE

The results of running unbalance optimization with optional cards confirmed the implication described above: the results are always equal or better when compared to the case with fixed cards. Yet, only 5/46 returned different values for max and mean unbalance. Two of them were able to slightly decrease the maximum unbalance ($\approx 1\%$), with an even lower increase of the mean unbalance. The other three led to improvements in the mean unbalance up to approximately 5%, without changes to the max unbalance. Remember that preemptive optimization is being used, so the value of the max unbalance could never be greater than the one obtained without optional cards.

The reduced number of card options considered in the test cases can be the main cause for the low improvement. Their choice does not considerably enlarge the possible combinations of loads' allocations regarding the different electrical phases.

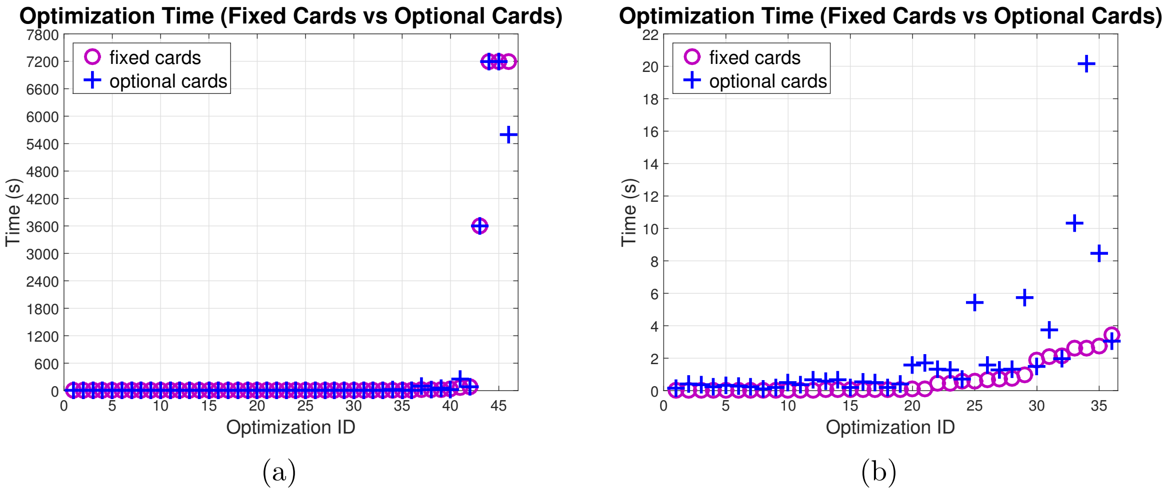


Figure 7.12: Running time for unbalance (preemptive) optimization with and without the possibility of changing cards (a) shows the time for completion for all allocation problems. (b) represents the same data, without the optimizations that reached the time limit.

The running times for the unbalance optimization are presented in Figure 7.12. The total time increases, with some exceptions. This is an expected behavior, since optional cards enlarge the set of possibilities.

Regarding the results of this optimization, special attention should be devoted to the choice of cards. If the unbalance optimization leads to a better balancing result than before (optimization with fixed system), it is acceptable that heavier cards were chosen or even added to the system. The optimization target is the unbalance and not the weight. However, if the result for unbalance is exactly the same with optional cards as it was with fixed cards, an increase in weight is not acceptable. The total results obtained, compared to the optimization with fixed system, were:

- 5/46 improved the unbalance values,
- 9/46 maintained the same unbalance, improving the weight of the system (4 of them even led to a lower number of cards),
- 21/46 maintained the same unbalance, keeping the system as it was with fixed cards, and
- 11/46 led to an heavier system (8 of them even increased the total number of cards).

Note that the 9/46 situations that led to an improved weight were merely a coincidence, since at this point no weight optimization is being done. There were 41/46 allocations with no improvement in the unbalance and 11 of them led to worse overall allocations.

This issue was predicted in Section 4.4.1 (see Figure 4.10) and the conclusion was that weight optimization should be performed at the end, in order to eliminate possible unnecessary cards or extra weight. This is done using preemptive optimization with fixed max and mean unbalances. The objective function is defined by the second target of weight optimization:

- Minimize the weight of the system, considering only cards. (Card Weight Optimization)

This was performed for every allocation. The final results obtained were (again comparing to the optimization with a fixed system):

- 5/46 improved the unbalance values,
- 19/46 maintained the same unbalance, improving the weight of the system (9 of them even led to a lower number of cards), and
- 22/46 maintained the same unbalance, keeping the system as it was with fixed cards.

Thus, this further optimization was able to completely eliminate the unacceptable allocations.

Obviously, this improvement has a cost: the extra running time. Most of them (42/46) finished within 10 seconds. There were two weight optimizations that considerably increased the running time. One of them took around 2200 sec, and the other was unable to finish before the time limit (3600 sec). Once more, these correspond to systems with a large number of variables. Nevertheless, they reached the final solution within the first 10 sec. Hence, when time limits are important, the running time for this optimization can eventually be diminished.

WEIGHT

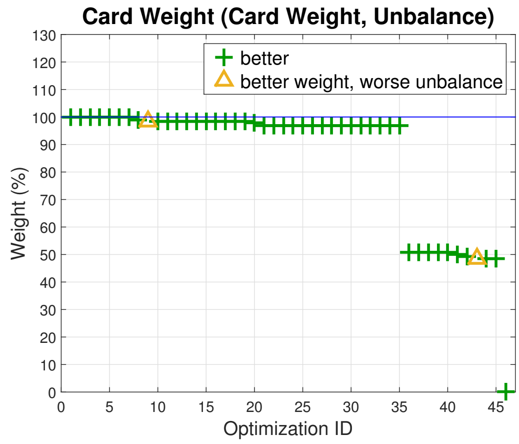


Figure 7.13: Weight minimization considering only cards. The optimization results are expressed in percentage with respect to the manual allocations.

Another possibility when optional cards are considered, is to first optimize the weight of the cards and then the unbalance. The results are presented in Figure 7.13. The weight is given in percentage, relative to the original manual allocation. Thus, optimizations located below the horizontal line (100%) describe improvements in terms of weight. Points on the line show situations where the weight was exactly the same before and after optimization. There are no points with weight over 100% (this would be considered as an error in the optimization program).

These results already include a further unbalance optimization (first max, then mean). This should always be done after any type of weight optimization. The reasons are analogous to the ones already given for the unbalance optimization. Without this step, we do not fully take advantage of the possible improvements in the overall allocation.

After these optimizations, it is possible to see in the figure that most of the situations led to improvements, both in weight and in phase balancing, when compared to the manual allocation. There are only two exceptions. However, this is acceptable, since the highest priority is assigned to weight.

Some of the new allocations (10/46) show reductions of about 50%. The rightmost one attains a weight reduction of 100%. This means that all the optional loads were allocated to cards containing standard loads. In terms of absolute value, the optimizations led to an average reduction of about 260 g per feeder (considering also the cases with no improvement). The reductions can go up to 2 kg per feeder.

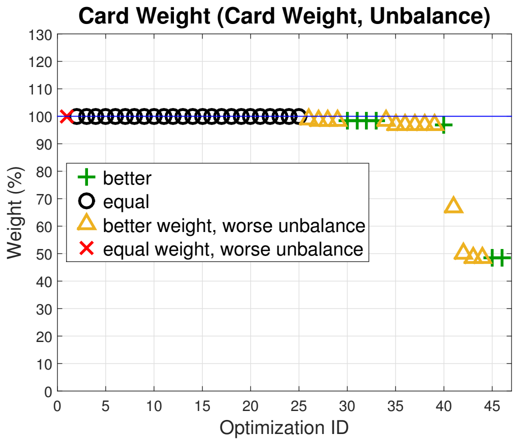


Figure 7.14: Weight minimization considering only cards. Comparison is now performed with respect to the optimization with inverse priority: first unbalance, then weight.

This comparison can also be done with the optimization with inverse priority (first unbalance, then weight). This is depicted in Fig. 7.14. Now there are more situations with better weight, but worse unbalance. This was predictable, since the prior optimization attained the best possible result in terms of unbalance. Some of these values cannot be achieved with the lighter system. Yet, there is a particular case that stands out from the rest, which has equal weight but worse unbalance. This is not a normal result. If the weight is the same, after the unbalance optimization, both allocations should yield the same values. The answer to this problem comes from the priorly defined stopping criterion. The difference is below $1 \cdot 10^{-4}\%$. This can be solved by diminishing the value of the stopping criterion.

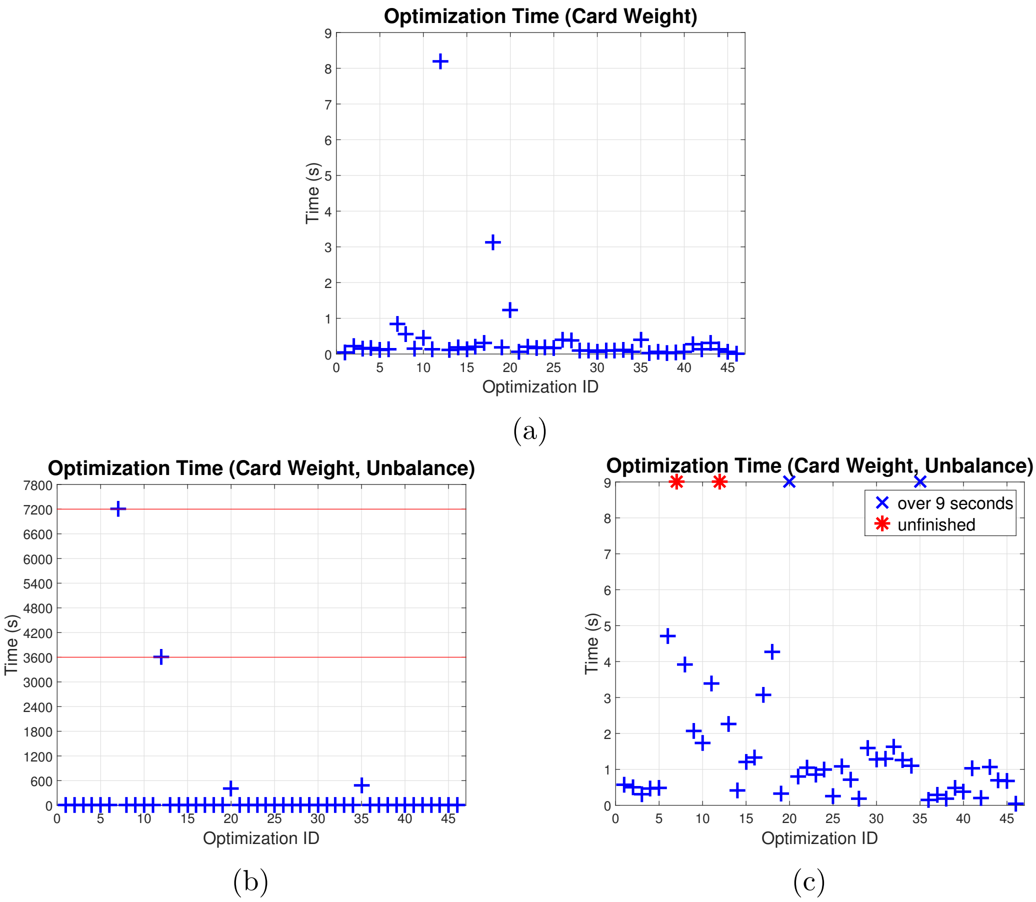


Figure 7.15: Running times for the weight optimization, considering only cards. Part (a) shows the time when single-objective optimization is used. Part (b) is the time with further unbalance optimization, and (c) represents the same data focusing on lower times.

The running times are also interesting (see Figure 7.15). In Fig. 7.15a, the times of the card weight optimization are shown. It is possible to see that this type of optimization has really good performance (in a user's point of view). All the optimizations were finished within approximately 8 seconds, and most of them even take less than a second.

When the subsequent optimizations (max and mean) are considered, the times increase substantially (see Figures 7.15b and 7.15c). Also, some optimizations reach the time limit.

When preemptive optimization is used, the increase in running time for the subsequent optimizations is a reasonable behavior. The lower number of feasible solutions, resulting from the extra constraints, can lead to additional difficulty in lowering the upper bound of the branch and bound (see Section 3.4.2). As a consequence, the number of fathomed nodes diminishes and the processing time increases. But this is certainly not the only reason. The problems with some of the unbalance optimizations were also observed when single objective was used (Figure 7.2a or 7.5a). So what is the difference between these optimizations?

One of the possible explanations can come from the number of optimal solutions. Consider the procedure for changing one load from a channel supplying phase A to another supplying phase B, in the same card. This change affects the value of unbalance, but does not affect the card weight. This is just an example, and the opposite could also be verified: if all the loads of one card option were changed to another card option, keeping the electrical phases, the weight would change and the unbalance would remain the same. Nevertheless, this is a less common event, and it is quite perceptible that the allocation decisions affect the values of unbalance more directly than the card weight. The same is to say that the card weight optimization should have more feasible solutions with the same value for the objective function, and particularly, for the same optimal value.

SUMMARY

Considering that cards can be changed, two main optimizations were performed: (1) preemptive optimization considering first the unbalance and then the weight of the cards, and (2) preemptive optimization considering the opposite order.

In the first case, no significant improvements in the unbalance were observed (only slight enhancements in 5/46). As already mentioned, the low number of card options can be a possible reason.

The second led to weight reductions that can go up to 2 kg per feeder (average is 260 g). When compared to the manual allocations, the unbalance was also improved in almost all of the situations (44/46). Comparing to the optimization with inverse priority, 21/46 allocations improved the weight. However, 14 of them were unable to attain the same result for the unbalance.

Regarding time, weight optimization is generally faster than unbalance optimizations. All of the card weight optimizations finished within 9 seconds. However, the further unbalance optimization boosts the completion time in some of the cases.

The optional cards affect the times of the unbalance optimization, but not in a user's point of view (a difference of 100% in one second, for example, is not significant for this type of problem).

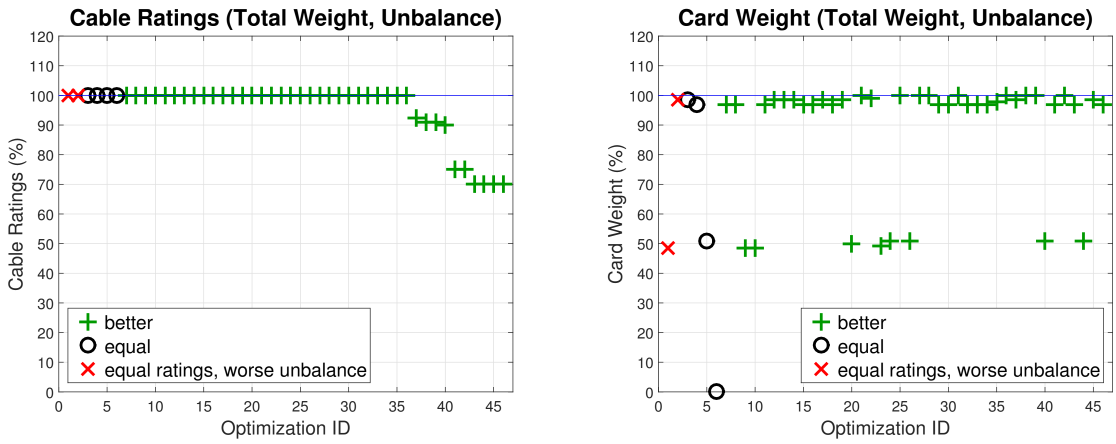
7.2.3 CUSTOMIZABLE SYSTEM

This section intends to present results for the situation where all system can be customized. That means that we are allowed to change cables, cards and the allocation of the optional loads. The results are going to focus preemptive optimization, with priority given to weight and then unbalance. The target of this weight optimization was defined in Section 4.2.1:

- Minimize the weight of the system, considering cables and cards.

The calculation of cable's weight is not straightforward. It depends on different parameters and on information not available within the scope of this work. In order to obtain results, some illustrative weight values were assigned to the different cable possibilities. The weights should be higher for the cable possibilities with larger current rating. However, when multiple cables are being optimized, the relations between these weights can also influence the choices. For example, suppose that there are three different possibilities for the cable ratings, $m_1 < m_2 < m_3$, with corresponding weights: $w_1 < w_2 < w_3$. Based on their relation, two cables m_2 can weigh more or less than two cables, one with rating m_1 and another with m_3. For the aim of this thesis, this is not an essential point, but it is something to keep in mind while running this type of optimization.

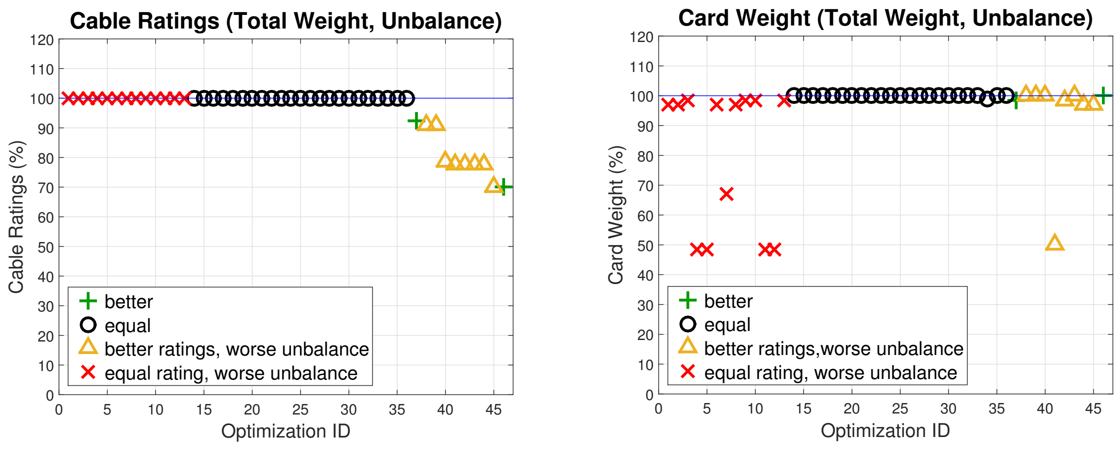
For the given reasons, possible improvements in cable weight are going to be measured in terms of cable ratings. As already stated during this thesis, the principle is: cables with lower ratings are usually thinner, leading to possible weight savings.



(a) (b)

Figure 7.16: Minimization of the total weight of the system (with further unbalance optimization). Part (a) shows the results for the cable ratings, as a percentage of the result for the manual allocation. Part (b) depicts the results for the card weight (also as a percentage).

The results are presented in Fig. 7.16. They represent the comparison between the optimized and the manual allocations for the test cases. From Fig. 7.16a, it is possible to verify that there is room for improvement in terms of cable ratings' reduction. This can probably lead to weight savings. Of the sets of test cases, 10/46 show margins for cable size reduction. This possible decrease in weight is accomplished with a diminished unbalance. So the program improves not only the total weight of the system, but also the other parameters. In 4/46 cases, the cable ratings and the unbalance are exactly the same before and after optimization. But, in two situations, no cable reduction is observed and the unbalance is worse. Note that keeping the cables with the same weight as before should lead at least to an allocation with the same quality, since preemptive optimization is used. Why does this happen? The fact is that the system is changed. The first optimization target considers the weight of cables and cards. Looking at Fig. 7.16b, there are improvements in the weight of the system, due to the change of cards. Particularly, the two situations on the left lead to a system with a better total weight value. Since unbalance optimization is only performed after obtaining the best weight for the system, these results are acceptable. It only shows that sometimes it is not possible to obtain the same result for unbalance, when a lighter system is chosen.

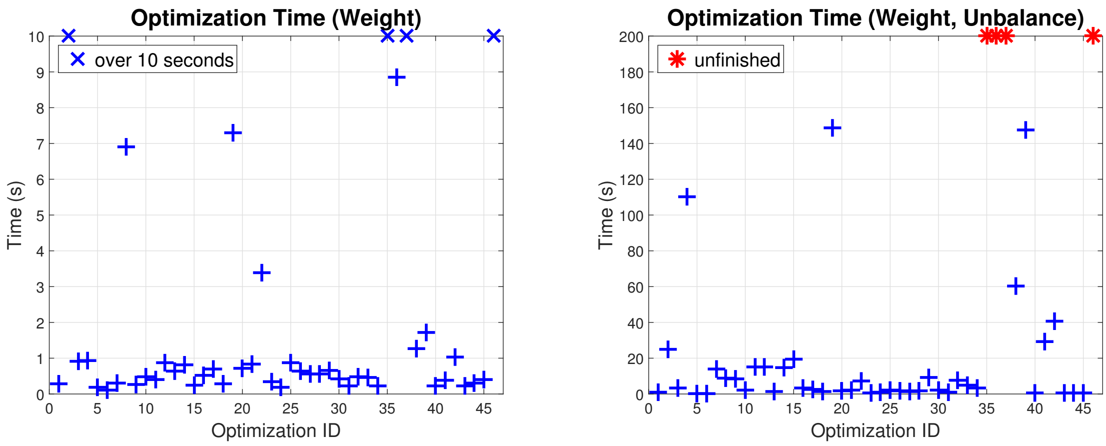


(a) (b)

Figure 7.17: Minimization of the total weight of the system (with further unbalance optimization). Comparison with optimization with inverse priority. Part (a) shows the results for the cable ratings, as a percentage of the result with inverse priority. Part (b) depicts the results for the card weight (also as a percentage).

The same analysis is done when comparing the results of this optimization with a situation where weight and unbalance are optimized, but in the opposite order. The results are presented in Fig 7.17. There are now situations where the system attains lower cable ratings but higher value of unbalance. Some of them are explained by different choices of cards. When priority is given to unbalance, the program is allowed to choose the cards that lead to the best possible value of unbalance. The weight is only subsequently reduced if the same value of unbalance can be achieved with lighter cables. So, it is normal that the optimization with weight priority possibly leads to worse values of unbalance, but with a lighter system.

Nevertheless, there are also situations which lead to the same card choices, but still result in different values of unbalance. The difference lies in the choice of the cables. This was also predicted in Section 4.4.1. The applicable limits must be verified for all electrical phases. So, at first sight, it may seem that keeping the power on the three electrical phases as balanced as possible would lead to the lowest possible cable ratings. In the referred section, some examples are given showing that this is not necessarily true when multiple feeders and/or boxes are considered together in the same optimization. There are four cases that perfectly illustrate this situation: same choice of cards, lower cable ratings and worse unbalance. This indicates that the optimization with priority given to unbalance was unable to return the best possible cable ratings.



(a) (b)

Figure 7.18: Running times for the weight optimization, considering a customizable system. Part (a) shows the time when single-objective optimization is used. Part (b) is the time with further unbalance optimization.

Regarding the optimization time, the results are similar to the case of card weight optimization. Figure 7.18a shows the optimization time exclusively for the weight optimization. It is comprehensible that it takes more time than the case where only the weight of the cards was considered. The choice of cables adds extra variables, which in many situations lead to an increased complexity. However, the running times are still very acceptable (when compared to the maximum time limit), and considerably faster when compared to the unbalance optimization. Most cases were solved within 10 seconds, and the most time-consuming took approximately 600 seconds. Also, in this situation, it is expected that for the same value of cable ratings many different allocations exist, with distinct values of phase balancing. Once again, this indicates that the optimal solution for weight is the same for many solutions of unbalance, leading to a faster optimization. The opposite is not verified.

The total time, considering also the unbalance optimization, grows considerably. Some situations, once again, reach the time limits. Three of them correspond to situations that also took more time (compared to average) to finish the single-objective weight optimization.

SUMMARY

Considering a customizable system, the optimization enables cable ratings' reduction in a significant number of situations (10/46). Some of them attain approximately 30% of reduction, when compared to the manual allocations. This can be an important result, when it comes to weight reduction in an aircraft. These results show that this tool may be able to help optimizing the aircraft's future network architectures.

One of the interesting features in weight optimization is the running time. No optimizations reached the time limits and most of them were finished within 10 seconds. This makes the tool very efficient in solving this type of problems.

CHAPTER 8

Tool Development

In this chapter, the implementation of the complete program is explained. The already implemented software to deal with customization was written in Microsoft® Excel 2010. The final version must then contain the necessary features so the optimization can be used together with the already implemented software. The resulting general requirements of the tool are:

- The optimization must be launched by clicking on a button inside the excel program.
- The system data must be read from the excel program.
- The load data must be read from the excel program.
- The optimization data must be defined inside the excel program.
- The program must give feedback if some of the parameters are not consistent.
- The program must give feedback during the optimization procedure.
- The result must be written inside the excel program.

In the following sections, each requirement is analyzed in more detail and the implemented solution for each of them is explained. At the end of the chapter, a general idea of the functional flow of the program is shown.

8.1 Launch Optimization

- **The optimization must be launched by clicking on a button inside the excel program.**

This procedure can be done in two steps: first create an application (.exe) from the MATLAB® code and then call the application from the excel file.

MATLAB® offers good solutions to create executables from a program, using the MATLAB® Compiler™ app. It offers the possibility of sharing programs as standalone applications that only use the MATLAB Compiler Runtime (MCR) , enabling royalty-free deployment to users who do not have MATLAB®. The MCR is a set of libraries that can be packaged together with the executable application, or downloaded during its installation.

With this method, the program can be developed and compiled into an executable file in a computer running MATLAB®, and then be used in any computer by simply launching the executable. This completely separates the processes of development and running the program, allowing a user completely unfamiliar with MATLAB® to use the application.

Since the optimization tool has to read and write all the data from and into the excel program, as it will be explained in a moment, the only necessary input for the application is the complete path of the excel file, and no output is needed. Hence, the main function can be created as:

```
function optimization(filename)
% OPTIMIZATION  Optimize the allocation of optional loads.
%   OPTIMIZATION(FILENAME) optimizes the allocation of the system defined in FILENAME.
%   FILENAME is the complete path of the excel file.
```

To run the optimization function passing the argument `filename` as input, the following command can be run from a DOS command prompt:

```
folderpath\optimization.exe filename
```

where `folderpath` is the path of the folder containing the executable file.

The call from the excel file can then be done by creating a button and assigning to it a macro written in Visual Basic for Applications (VBA) . This macro can call the function in the following way:

```
Sub optimizationCall()
  Dim appname As String
  Dim filename As String
  appname = "folderpath\optimization.exe "   'path to the executable
  filename = ActiveWorkbook.FullName   'full path to the current excel file
  Shell(appname & filename)   'run the application
End Sub
```

HINTS

Input arguments. The input arguments to an executable generated by the MATLAB® Compiler are automatically parsed, using space as the delimiter. This means that the following command is used to pass multiple inputs:

```
folderpath\optimization.exe argument1 argument2 ...
```

When the string containing the path to the excel file is passed as argument to the executable, a problem occurs if the path contains spaces. This leads to multiple input arguments, instead of only one. A possible solution is to include quotation marks to delimit the string. The executable interprets strings delimited by quotation marks as a single argument, even if spaces are present. Therefore a simple change can be made to the above VBA code:

```
filename = char(34) & ActiveWorkbook.FullName & char(34)
```

where `char(34)` is the character code for quotation marks.

Compiler. Luckily, all toolboxes and functions used in the development of this MATLAB® program are supported by the current compiler. But this is not always the case. Although it supports most of the toolboxes and functions, when starting a complex project it is probably a good idea to check if the most important functions for the project are supported by the compiler, specially those that involve a lot of manpower to develop (for example `intlinprog`).

Another important directive related to compiling and running a MATLAB® application is to ensure that the MCR matches exactly the version that was used to develop the program. There is an MCR for each new MATLAB® version.

8.2 READ SYSTEM DATA

- **The system data must be read from the excel program.**

System data includes the following:

- feeder data,
- box (SPDB) data, and
- card (LRM) data.

Feeder data refers to the number of feeders and their designation, the boxes that each of them supplies, and the power limits of each of the segments.

Box data includes the number of card slots that each box contains and the feeder that supplies each of these card slots.

Card data is the type of card for each slot, which defines the number of channels and their ratings. To enable more flexibility in the customization process, an extra sheet was created to allow the user to choose which of the card slots can be considered as optional.

MATLAB® provides a function to read from excel, with the following syntax:

```
[num,txt,raw]=xlsread(filename,sheet)
```

the input parameters, as their names indicate, are: the name of the excel file (complete path) and the name of the sheet to be read. The output is divided in `num`, `text` and `raw`. The variable `num` stores the numerical values of the sheet, `txt` the text entries, while `raw` stores every entry in a cell structure that can then be converted to number or text.

8.3 READ LOAD DATA

- **The load data must be read from the excel program.**

Load data is the information that characterizes each load. This is similar to the data presented in Table 2.1. In a normal situation, this information already contains the optional/standard load specification. However, an extra sheet was created to allow the user to customize this definition for each load. Special care must be taken if an originally optional load is considered as standard, since its position must then be read from a previous allocation scheme.

8.4 DEFINE OPTIMIZATION DATA

- **The optimization data must be defined inside the excel program.**

The definition of optimization data consists in the following:

- choice of feeders/boxes (SPDBs) to optimize,

- optimization targets,
- weight of the different card options,
- rating of the different cable options and their weight,
- importance weight for the different flight phases, and
- maximum time per optimization.

The first two have some particularities, so they will be explained in more detail. The handling of the remaining is relatively straightforward: their values are read and attributed to the respective variables.

8.4.1 FEEDERS/BOXES TO OPTIMIZE

FEEDERS

As previously described, the optimization is performed on the feeder level. But this is not as simple as it may seem at first. Consider again the example of Figure 2.9. When optimizing feeder 22AC, feeder 26AC must be optimized at the same time, since they are both affected by the allocation of the same group of loads. If feeder 22AC is optimized alone, there is a perfect solution for every target: just allocate all loads to channels supplied by feeder 26AC. This leads to zero unbalance in feeder 22AC, all the cables can be as thin as we can imagine, and all the channels remain free for further allocations. Then feeder 26AC is optimized and the opposite solution is obtained: allocate all loads to channels supplied by feeder 22AC. Clearly, feeders 22AC and 26AC should be optimized together.

Now look at the example given in Figure 2.6. Suppose the idea is to optimize the whole system of side 2, i.e. all the feeders with a pair number designation (only AC are considered now). How do we know which of them have to be optimized together? Consider the optimization of feeder 20AC. This feeder supplies box number 2. However, box 2 is also supplied by feeder 24AC, so this feeder should also be optimized together with 20AC. Still, to optimize feeder 24AC, all the boxes supplied by it must be taken into account. Since it supplies boxes 2 and 6, the feeders that supply these should also be considered. This suggests a recursive procedure.

First, let us define the function:

```
function B = getFeederBoxes(feederdata,F)
% GETFEEDERBOXES  Returns the boxes that are supplied by each feeder.
%   B = GETFEEDERBOXES(FEEDERDATA,F) returns the
%   matrix B, whose lines correspond to the feeders
%   contained in the F vector. For every line the matrix contains
%   the numbers of the boxes that are supplied by that feeder. FEEDERDATA
%   is the structure containing the information of the feeders.
```

Considering `F` as the vector with the feeders of the system:

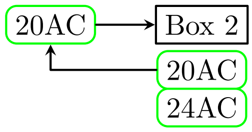
$$\texttt{F} = \begin{pmatrix} 20\text{AC} & 22\text{AC} & 24\text{AC} & 26\text{AC} & 28\text{AC} \end{pmatrix}, \tag{8.1}$$

the output `B` is then given by (see Figure 2.6):

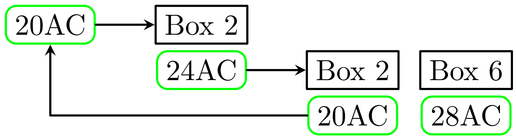
$$\texttt{B} = \begin{array}{c} 20\text{AC} \\ 22\text{AC} \\ 24\text{AC} \\ 26\text{AC} \\ 28\text{AC} \end{array} \begin{pmatrix} 2 & 0 \\ 4 & 8 \\ 2 & 6 \\ 4 & 8 \\ 6 & 0 \end{pmatrix}, \tag{8.2}$$

where 0 is the same as no box.

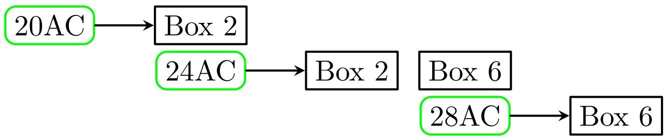
Recap the previous example. Start at feeder 20AC and find the boxes that it supplies. The answer is box 2. Now find the feeders supplying box 2. The result are feeders 20AC and 24AC. If we start the procedure again for the feeders found, we can end up in an infinite loop with feeder 20AC. This is illustrated in the following diagram:



A possible turnaround may be to simply consider the feeders supplying box 2 excluding the starting feeder. This solves the previous problem. However, when the procedure is repeated for 24AC, this last feeder is excluded from the search, but feeder 20AC reappears, leading again to an infinite loop:



The solution is to eliminate the analyzed feeder from all the subsequent searches:



But before eliminating the feeders from future searches (which means to set all values to zero in the corresponding row of matrix `B`), there must be a structure keeping track of the already considered feeders, since they are going to belong to the same optimization problem. For this purpose, a vector containing the optimization number (initially composed with zeros) is defined:

$$\texttt{optn} = \begin{array}{c} \begin{array}{ccccc} \text{20AC} & \text{22AC} & \text{24AC} & \text{26AC} & \text{28AC} \end{array} \\ \begin{pmatrix} 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 \end{pmatrix} \end{array}. \tag{8.3}$$

This vector is updated by considering all the feeders found for each inspection of feeder dependencies. The following function was developed to search for the relations between feeders:

```
1 n = 1;   % index for optimization number
2 for f=1:feedernum   % for every feeder
3   if(sum(boxtotal(f,:)~=0))   % feeder supplies at least one box
4     % call recursive function to search for dependencies
5     [optn,B] = searchDependencies(optn,B,f,n);
6     n=n+1;   % increment the optimization number
7   end
8 end
```

Note that the recursive procedure is called for each feeder. However, `B` is updated during this process (it is both an input and an output). When a given feeder appears in one of the calls to `searchDependencies`, all the elements in `B` belonging to that feeder line are set to 0. Then, this feeder will not verify the condition of line 3, and the function will not be called. In the above example, the first feeder is the 20AC. After calling `searchDependencies` for this feeder, 24AC and 28AC will be identified as depending on it, and the outputs will be:

$$\texttt{optn} = \begin{array}{c} \begin{array}{ccccc} \text{20AC} & \text{22AC} & \text{24AC} & \text{26AC} & \text{28AC} \end{array} \\ \begin{pmatrix} 1 & \quad 0 & \quad 1 & \quad 0 & \quad 1 \end{pmatrix} \end{array} \tag{8.4}$$

$$
\mathtt{B} = \begin{array}{c} 20\text{AC} \\ 22\text{AC} \\ 24\text{AC} \\ 26\text{AC} \\ 28\text{AC} \end{array} \begin{pmatrix} 0 & 0 \\ 4 & 8 \\ 0 & 0 \\ 4 & 8 \\ 0 & 0 \end{pmatrix}. \tag{8.5}
$$

This ensures that there is only one optimization number assigned to each feeder.

The function `searchDependencies` is explained by the flow chart shown on the next page.

The final result is an optimization number assigned to every feeder. Feeders that must be optimized together carry the same optimization number. In the example considered:

$$
\mathtt{optn} = \begin{array}{c} \\ \big(\end{array} \begin{array}{ccccc} 20\text{AC} & 22\text{AC} & 24\text{AC} & 26\text{AC} & 28\text{AC} \\ 1 & 2 & 1 & 2 & 1 \end{array} \begin{array}{c} \\ \big). \end{array} \tag{8.6}
$$

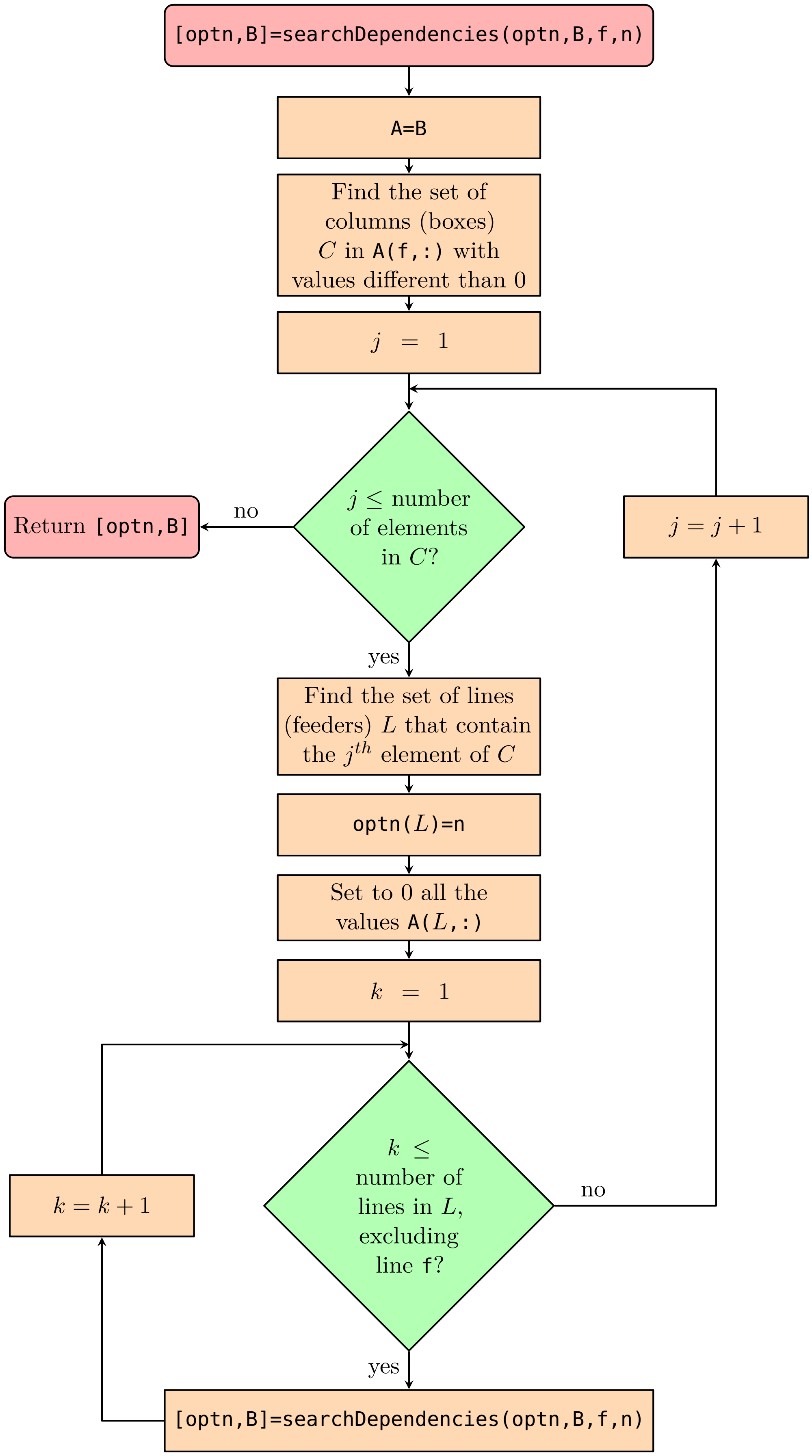


Figure 8.1: Flow chart of the function to search for dependencies between feeders.

BOXES

The boxes to optimize are affected by a similar problem. Consider, from the example of Figure 2.9, that we want to optimize SPDB 4. The calculations for the optimization of the different feeders (22AC, 26AC and 24DC) are not only dependent on the allocation decisions inside this box, but also on the load distribution inside SPDB 8. However, in this situation, it is assumed that the user wants to keep the allocation scheme in box SPDB 8 as it is, only allowing changes in SPDB 4. This problem can be solved using the optional/standard load definition, by considering all the loads in SPDB 8 as standard. Still, this case is quite simple, since all the feeders supplying SPDB 4 only supply these two boxes. However, with a structure like the one depicted in Figure 2.6, this becomes more complicated. The procedure to identify the affected boxes and the corresponding loads is performed with the following steps:

1. Run `getFeederBoxes` with the input `F` containing all feeders of the system. The output is the matrix `B` with all the boxes supplied by each of the feeders.
2. For every chosen box for the optimization $b \in B_{\text{optim}}$, find the feeders supplying it. This means identifying the lines in `B` that contain at least one of the $b \in B_{\text{optim}}$. These feeders make the set F_{optim} (set of feeders to be optimized).
3. Find the set of boxes B' that are supplied by all of the feeders F_{optim} found in step 2, by running `getFeederBoxes` with the input `F` composed by all feeders in F_{optim}. Note that the chosen boxes B_{optim} are part of the set B'.
4. All loads belonging to the set of boxes B' found on the previous step, excluding those chosen for optimization (B_{optim}), are considered as standard loads.

8.4.2 OPTIMIZATION TARGETS

In the developed tool, the optimizations can be performed with any of the available targets, by choosing them inside the excel file. It allows single-objective and multi-objective optimization. When multiple targets are chosen, preemptive optimization is done based on the priority assigned by the user. Hence, multi-objective preemptive optimization can be performed with any combination of targets and priorities.

As it was mentioned during the introduction to preemptive optimization (Section 3.5.1), for each subsequent problem with a new target, the results from the previous optimizations are formulated as constraints, using the corresponding objective function. This extends the number of constraints and possibly the number of variables (depending on the objective function). Regarding the MATLAB® syntax for the `intlinprog`, this can have direct impact on all of the input variables.

Consider the following function example to optimize the phase balancing (Target 2: Minimize the mean unbalance among all feeders and all flight phases):

```
function [x,val] = phaseMeanOptimization(system,opt,std,targets)
% PHASEMAXOPTIMIZATION Create the necessary structures and run intlinprog
% for optimizing the maximum unbalance among all feeders.
%   [X,VAL] = PHASEOPTIMIZATIONMEAN(SYSTEM,OPT,STD,TARGETS) optimizes the average
%   unbalance among all feeders and flight phases. It returns the vector X, containing the value for
%   each of the variables, and the optimization value VAL. SYSTEM is the structure containing
%   all the information of the system to optimize. OPT and STD are structures with
%   all the necessary information of optional and standard loads,
%   respectively. TARGETS is a structure with the values for each of the
%   former optimizations.
```

The key structure for preemptive optimization is defined by `targets`. This variable contains one entrance for each of the possible target choices:

```
targets.maxunb
targets.meanunb
...
targets.cardweight
...
```

Initially all the variables from this structure are empty. The values are changed when an optimization is performed. Consider that there was a prior optimization regarding card weight, and that the result W_1 was obtained. Then,

`targets.cardweight` = W_1.

Inside each optimization function, such as `phaseMeanOptimization`, there are essentially six steps, directly connected to the inputs for the `intlinprog`. These are:

- Define the variables' indexes.
- Define which of the variables are integer (input `intcon`).
- Define the boundaries for each variable (inputs `ub` and `lb`).
- Define the constraints formulated as inequalities (inputs `A` and `b`).
- Define the constraints formulated as equalities (inputs `Aeq` and `beq`).
- Define the objective function (input `f`).

Defining the variables' indexes is necessary for the construction of all the matrices and vectors, since they must be consistent throughout the problem definition. For this purpose a structure named `varindex` is defined inside each optimization function. In this example, it contains the different indexes for the variables $x_{i,j}$ and also for the variables $U_{\gamma,f}$ (see (5.21)). Hence,

`x_num` = $N_L \cdot N_S$
`U_num` = $N_\Gamma \cdot N_F$
```
N = x_num + U_num
varindex.x_i = 1
varindex.U_i = varindex.x_i + x_num
varindex.x_f = varindex.U_i−1
varindex.U_f = N
```

where N_L and N_S are the number of loads and channels, respectively. N_Γ is the number of flight phases and N_f the number of feeders. The value `x_num` represents the number of $x_{l,s}$ variables, and `U_num` is the number of variables $U_{\gamma,f}$. Thus `N` is the total number of variables.

All the additional definitions needed, for taking into account a prior optimization on card weight, are obtained by calling the following function:

```
function [varargout] = addCardWeightOpt(varargin)
```

where `varargin` and `varargout` mean that it has a variable number of input and output arguments. The usage of this function is based on the first input argument, which is a string identifying the situation:

- `'var'` - necessary changes to the `varindex` structure by adding the needed variables. The total number of variables `N` is also updated.
- `'intvar'` - updates the variable `intcon` with the additional integer indexes (see Chapter 6 for the explanation of the inputs of `intlinprog`).

- `'ineqnum'` - updates the number of inequalities, by summing the number of inequalities arising from the card weight optimization. This step is important for the reasons presented in Chapter 6. The performance of MATLAB® strongly improves when the necessary memory is priorly allocated. Therefore, updating this value enables the creation of matrix `A` and vector `b` with the proper size.
- `'ineq'` - updates the matrix `A` and the vector `b` with the additional inequalities coming from the card weight formulation. Note here that a variable `index` should be passed as input argument and should be updated inside the function. This helps to keep track of the index of the next empty line to write constraints.
- `'ineqval'` - updates matrix `A` with the inequality responsible for the fulfillment of the prior card weight optimization. Note that one of the input parameters should be the value `targets.weightcard`.
- `'eqnum'` and `'eq'` are analogous to `'ineqnum'` and `'ineq'`, but for the equalities.

A draft of the different calls is given below:

```
[N,varindex] = addCardWeightOpt('var',N,varindex,...);
[intcon] = addCardWeightOpt('intvar',intcon,varindex,...);
[ineqnum] = addCardWeightOpt('ineqnum',ineqnum,...);
[A,b,index] = addCardWeightOpt('ineq',A,b,index,varindex,...);
[A,b,index] = addCardWeightOpt('ineqval',targets.weightcard,A,b,index,varindex,...);
[eqnum] = addCardWeightOpt('eqnum',eqnum,...);
[Aeq,beq,index] = addCardWeightOpt('eq',Aeq,beq,index,varindex,...);
```

Both functions presented (`phaseMeanOptimization` and `addCardWeightOpt`) serve merely as an example to show that for each of the optimization targets there are two main functions: (1) the function responsible for optimizing a certain target, and (2) the function to include all the necessary changes, in order to consider the prior optimization of that target. This type of organization enables any combination of prior optimizations to be taken into account.

HINT

When preemptive optimization is performed, as described in section 3.5.1, we must add constraints of the form:

$$f_j(x) \leq y_j^* \tag{8.7}$$

where $f_j(x)$ is the objective function of a previous optimization and y_j^* is the objective function's value for the solution obtained. When working with computers, since memory is not unlimited, rounding or truncating is inevitable. This is one good reason to always consider an inequality (lower or equal), instead of an equality. Moreover, rounding and truncating can lead to situations where the presented value y_j^* is slightly lower than the true output. When performing further optimizations with this value as a constraint, it can happen that no feasible points are found, due to the impossibility of fulfilling this constraint. A possible solution is to sum a small amount to y_j^*. It should be small enough to ensure that no solutions lie in the interval between y_j^* and the new value.

8.5 INCONSISTENCY IN PARAMETERS

- **The program must give feedback if some of the parameters are not consistent.**

This analysis is performed considering two main sources of inconsistency:

- system data, and
- optimization data.

Inconsistency in system data involves problems in the definition of the various characteristics of the system. A simple example of this can be having more loads than channels for allocation. Inconsistency in optimization data is related to problems regarding the choices of the user, such as trying to optimize a single feeder when the allocations affect simultaneously multiple feeders.

8.5.1 SYSTEM DATA

The problems regarding system data are easily detected but quite hard to identify. The detection is done automatically when no feasible points are found, meaning that the problem has no solution. This is immediately identified by `intlinprog`, returning the following message:

```
No feasible solution found.

Intlinprog stopped because no point satisfies the constraints.
```

Now, the person using the software asks: "Okay, no feasible solution... What to do next?". Although fast, this feedback is quite poor in a user's point of view. A possible solution can be to try to identify which constraints cannot be verified. Hence, the following process could be used:

1. Change maximum `intlinprog` running time to zero. This means that only preprocessing is made (such as LP relaxation), branch and bound is not started. But this is enough to check for feasibility.
2. Eliminate a constraint (such as the applicable limits coming from power limitations).
3. Run `intlinprog` with the new formulation.
4. If no feasible solutions are found, restore the constraint and return to point 2 (eliminate another constraint).

This process can also eventually be done for groups of constraints, if this first attempt does not solve the problem.

The order of the constraints' elimination is important. For example, consider that there is no way to satisfy the applicable limits with the available possible connections. If we then consider that all loads can be connected to any current rating and arbitrary electrical phase (eliminating Feasibility Matrix constraints), the new allocation may be able to fulfill the applicable limits. This can wrongly lead to the conclusion that the issue lies in the types of loads and channels, when in fact the problem was the size of the cables.

8.5.2 OPTIMIZATION DATA

As previously indicated in section 8.4.1, the user is able to choose the feeders/boxes that are going to be optimized. However, it was shown that some feeders should be optimized together. Therefore, if only some of the feeders, belonging to a single optimization, are chosen, the program does not perform the optimization and gives back a warning. This warning explains the situation to the user, identifying the feeders that should also be considered. For the example given in the referred section, suppose that only feeder 20AC was chosen for optimization. Then the tool would give the following warning:

```
Warning: To optimize feeder 20AC, feeders 24AC and 28AC should also be considered.
```

Another situation involves the optional cards. The user is able to choose which of the card slots are optional. A problem occurs if the user allows a card to be changed, when it has some standard loads allocated to it. To solve this issue, the tool reads all the standard loads in the beginning and identifies the situations where an optional card has at least one standard load. This card is then considered as fixed and the following warning is given (example):

```
Warning: Card 1 from box 3 has standard loads allocated to it, so it is going to be
considered as a fixed card.
```

8.6 FEEDBACK DURING OPERATION

- **The program must give feedback during the optimization procedure**

The function `intlinprog` already incorporates an iterative display. This is a table of statistics describing the iterations of the solver. This function, in particular, shows for each iteration the number of nodes explored, the total time (in seconds) since it started, the number of integer solutions found, the objective function value for the best integer solution found, and finally, the relative gap (explained during the analysis of the results).

The iterative display can be included in the executable file. MATLAB® Compiler gives the possibility of displaying this information during the execution of the program, using a console window. Therefore all the warnings and outputs from the program (including those from `intlinprog`) can be visualized in real time. Moreover, a file containing this output can be saved each time the tool runs. This is important to allow further inspection.

8.7 WRITE OUTPUT

- **The result must be written inside the excel program**

INFORMATION TO WRITE

For every optimization performed, there are essentially two main outputs: (1) the values of the decision variables and (2) the value of the objective function. The first determines the allocation chosen and it must be written inside the excel file. The second determines the quality of the solution. Nevertheless, since multiple optimization targets exist, it is important for the user to have the possibility of comparing two allocations that were not optimized according to the same target. Due to this, for each optimization, the results for every objective function are calculated and written inside the excel file.

HOW TO WRITE

MATLAB® provides a function to write to an excel file, with the following syntax:

```
xlswrite(filename,A,sheet,xlRange)
```

As the name indicates, `filename` and `sheet` are the name of the excel file and the sheet where information is going to be written, respectively. `A` is the array with the information to be written, and `xlRange` is the range of cells where the information should be written.

Unfortunately, this function does not enable writing into open files. In the developed tool, it is convenient that the information can be written without having to close the file every time an optimization is performed. There is a workaround, by getting a handle to the excel application. After this, writing in excel can be performed while the file is open, using VBA code. The following code performs the operation of writing an array `A` to the sheet `sheetname` inside the excel file `filename`. `cells` is the range of cells where the information should be written (for example, `cells = 'B2:C10'`):

```
%Get handle to the open Excel
h = actxGetRunningServer('Excel.Application');
%Get workbook handle
myBook = h.Workbooks.Item(filename);
%Get sheet handle
mySheet = myBook.Sheets.Item(sheetname);
%Write values
mySheet.Range(cells).Value = A;
```

8.8 FUNCTIONAL FLOW

The general flow chart of the complete tool is depicted in Figure 8.2. The tool starts by reading all system data to identify the feeders/boxes to be optimized and also the number of optimization problems. This was explained in Section 8.4.1. All data from standard loads is also read to identify possible situations of inconsistency (see Section 8.5.2).

For each optimization problem, the system and load data are read. Based on it, the power supplied to electrical loads is calculated for each segment. This is necessary for the definition of constraints. The Feasibility Matrix (defined in Chapter 6) is also created.

There is a point not previously mentioned. Some optimizations may take considerable time (this was observed during the analysis of the results). Consider that a user already has an optimized allocation for electrical phase unbalance. If there is the need to obtain an allocation with optimized values of unbalance and weight (with this order of priority), the program would run unbalance optimization again and then include this result as a constraint to the weight optimization. This would lead to extra unnecessary time. Therefore, the user is able to introduce the values of prior optimizations as input, and they are read before starting the optimization functions.

For each optimization, the process explained in Section 8.4.2 is performed. When all the targets from the preemptive optimization are optimized, the tool writes the output to the excel program, and starts the next optimization problem.

The verification of the validity of the solutions found was already available, and is done outside the developed tool.

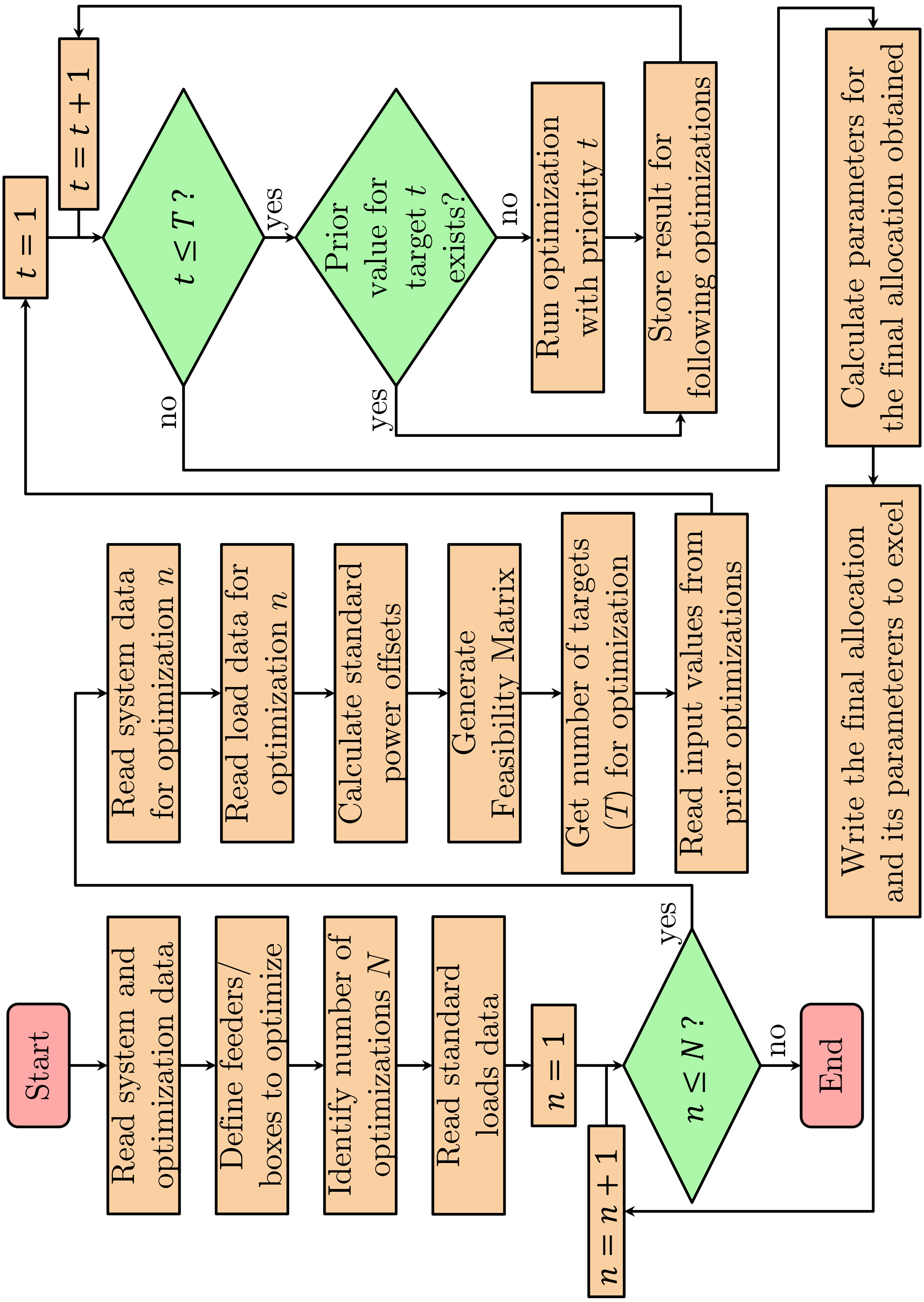


Figure 8.2: General flow chart of the complete tool.

CHAPTER 9

Conclusion

During this work, an example of an aircraft's cabin and cargo electrical system was studied, and scalable mathematical models were developed, in order to cover many different scenarios. In addition, a complete tool to automate the allocation procedure was developed. This tool enables the optimization of the allocation problems, according to different targets with single-objective or multi-objective optimization (mainly based on the preemptive method). Furthermore, it was developed to be completely integrated with the previously available software.

In this thesis, two main criteria were used for optimizing the cabin and cargo connection decisions: (1) three-phase unbalance and (2) weight. The first criterion is important when considering a fixed system, since in these cases there is no possibility of weight savings. Unbalance between the electrical phases can have negative impacts on the power distribution system, so it is important to keep it as low as possible. The second becomes the most important target when changes in the system are allowed.

Different allocation scenarios were tested, and the results were validated using the previously available software. According to the chosen definition for unbalance calculation, the developed tool was able to significantly reduce the unbalance in most of the test cases, when compared to the manual allocation. However, these results must be regarded with special care. Due to the unavailability of the power factor's angle information, the calculations of unbalance may not be accurate in some situations. The software can be adapted to include this information in the future.

Substantial research is devoted to weight improvements in aircrafts, as it is one of the possible ways to increase the profit of airline companies. Some of the results obtained for weight optimization enable cable rating reductions, which can eventually be used to lower the corresponding system's weight. Therefore, this is one of the main outcomes of this thesis. Optimized cabin and cargo allocations can be a way to enhance the aircraft efficiency.

Besides leading to improved results, the software is also able to significantly lower the necessary time for cabin and cargo connection decisions. Even when a non-fixed system is used, the optimization regarding the multiple targets is completed within a few minutes, in most of the tests cases. Some cases with a high number of loads lead to unfinished optimizations (according to the time limits established). Still, in these cases, the results obtained after running the tool for even just a few minutes already show considerable enhancements. The high complexity associated with these cases (manifested through the running time) poses even greater difficulties to the manual allocation. Thus, they are normally associated with possible high improvements. In summary, the tool is considerably faster than the manual procedure and may result in substantial improvements in the total customization time.

Finally, although this thesis focuses on cabin and cargo electrical power systems, the studies and software developed can be adapted to consider other parts of the aircraft's power system, or any system containing similar architectures.

References

[1] C. K. Alexander and M. N. O. Sadiku, *Fundamentals of Electric Circuits, 2nd Edition*, 2nd. Mcgraw-Hill (Tx), Dec. 2002, ISBN: 9780072463316.

[2] M. Anwari and A. Hiendro, «New unbalance factor for estimating performance of a three-phase induction motor with under-and overvoltage unbalance», *Energy Conversion, IEEE Transactions on*, vol. 25, no. 3, pp. 619–625, 2010.

[3] Armed Forces International. (2013). Rp-2032151xd0 and rp-26231000n1 solid-state power controllers, [Online]. Available: `http://www.armedforces-int.com/article/solid-state-power-controllers.html` (visited on 10/15/2014).

[4] T.-H. Chen, «Evaluation of line loss under load unbalance using the complex unbalance factor», *IEE Proceedings-Generation, Transmission and Distribution*, vol. 142, no. 2, pp. 173–178, 1995.

[5] M. Chindris, A. Cziker, A. Miron, H. Balan, A. Iacob, and A. Sudria, «Propagation of unbalance in electric power systems», in *Electrical Power Quality and Utilisation, 2007. EPQU 2007. 9th International Conference on*, IEEE, 2007, pp. 1–5.

[6] I. Christou, A. Nelms, I. Cotton, and M. Husband, «Choice of optimal voltage for more electric aircraft wiring systems», *Electrical Systems in Transportation, IET*, vol. 1, no. 1, pp. 24–30, 2011.

[7] D. Colombo, A. Lamantia, and T. Stock, «Solid state power controller (sspc): an application perspective», 2012. [Online]. Available: `https://www.see.asso.fr/file/3179/download/5619`.

[8] I. Cotton, A. Nelms, and M. Husband, «Defining safe operating voltages for aerospace electrical systems», in *Electrical Insulation Conference and Electrical Manufacturing Expo, 2007*, IEEE, 2007, pp. 67–71.

[9] I. Cotton and A. Nelms, «Higher voltage aircraft power systems», *Aerospace and Electronic Systems Magazine, IEEE*, vol. 23, no. 2, pp. 25–32, 2008.

[10] E. Danna, E. Rothberg, and C. Le Pape, «Exploring relaxation induced neighborhoods to improve mip solutions», *Mathematical Programming*, vol. 102, no. 1, pp. 71–90, 2005.

[11] R. C. Dorf and J. A. Svoboda, *Introduction to Electric Circuits*, 7th ed. Wiley, Jan. 2006, ISBN: 9780471730422.

[12] M. Fischetti and A. Lodi, «Local branching», *Mathematical programming*, vol. 98, no. 1-3, pp. 23–47, 2003.

[13] S. I. Gass and A. A. Assad, *An Annotated Timeline of Operations Research: An Informal History*. Springer, 2005, vol. 75.

[14] J. J. Grainger and W. D. Stevenson, *Power system analysis*. McGraw-Hill New York, 1994, vol. 621.

[15] N. Hadjsaid and J.-C. Sabonnadiere, *Power systems and restructuring*. John Wiley & Sons, 2013.

[16] F. Hillier and G. Lieberman, *Introduction to Operations Research*, ser. Introduction to Operations Research. McGraw-Hill Higher Education, 2010, ISBN: 9780071267670. [Online]. Available: `http://books.google.de/books?id=NvE5PgAACAAJ`.

[17] A. Kalyuzhny and G. Kushnir, «Analysis of current unbalance in transmission systems with short lines», *Power Delivery, IEEE Transactions on*, vol. 22, no. 2, pp. 1040–1048, 2007.

[18] Leach International. (2003). Solid-state power controllers, [Online]. Available: `http://www.esterline.com/Portals/3/Literature/Solid%20State%20Power%20Controllers.pdf` (visited on 10/15/2014).

[19] Z. Liu and J. Milanovic, «Probabilistic estimation of propagation of unbalance in distribution network with asymmetrical loads», 2012.

[20] M. Maasoumy, P. Nuzzo, F. Iandola, M. Kamgarpour, A. Sangiovanni-Vincentelli, and C. Tomlin, «Optimal load management system for aircraft electric power distribution», in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, IEEE, 2013, pp. 2939–2945.

[21] MathWorks. (2014). Full and sparse matrices, [Online]. Available: `http://www.mathworks.com/help/matlab/math/full-and-sparse-matrices.html` (visited on 08/14/2014).

[22] ——, (2014). Mixed-integer linear programming algorithms, [Online]. Available: `http://www.mathworks.com/help/optim/ug/mixed-integer-linear-programming-algorithms.html` (visited on 09/09/2014).

[23] ——, (2014). Techniques for improving performance, [Online]. Available: `http://www.mathworks.com/help/matlab/matlab_prog/techniques-for-improving-performance.html` (visited on 10/03/2014).

[24] I. Moir and A. Seabridge, *Aircraft Systems: Mechanical, Electrical and Avionics Subsystems Integration*, 3rd ed. Wiley, May 2008, ISBN: 9780470059968.

[25] A. S. Nautiyal, B. P. Thakur, C. A. Chauhan, and D. K. Govind, «Estimating torque-speed characteristic of three-phase induction motor operating under unbalance supply», in *Engineering (NUiCONE), 2013 Nirma University International Conference on*, IEEE, 2013, pp. 1–6.

[26] NEMA, *Motors and Generators, Part 14.36: Effects of Unbalanced Voltages on the Performance of Polyphase Induction Motors*. NEMA Standard MG1-1998 (Revision3, 2002), 2002.

[27] J. W. Nilsson and S. Riedel, *Electric Circuits (9th Edition)*, 9th ed. Prentice Hall, Jan. 2010, ISBN: 9780136114994.

[28] B. Nya and D. Schulz, «Approach to implementation of power management with load prioritizations in modern civil aircraft», in *Proceedings of World Academy of Science, Engineering and Technology*, World Academy of Science, Engineering and Technology, 2012.

[29] R. L. Rardin, *Optimization in operations research*. Prentice Hall New Jersey, 1998, vol. 166.

[30] D. Schlabe and J. Lienig, «Energy management of aircraft electrical systems-state of the art and further directions», in *Electrical Systems for Aircraft, Railway and Ship Propulsion (ESARS), 2012*, IEEE, 2012, pp. 1–6.

[31] T. Schröter, *Power Management on Aircraft*. VDE Verlag, 2013.

[32] T. Schröter and D. Schulz, «The electrical aircraft network—benefits and drawbacks of modifications», *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 1, pp. 189–200, 2013.

[33] Sensitron. (2013). Solid state power management, [Online]. Available: `http://sensitron.com/Hi_brochure/SolidStateSolutions.pdf` (visited on 10/15/2014).

[34] J. C. Smith and Z. C. Taskin, «A tutorial guide to mixed-integer programming models and solution techniques», *Optimization in Medicine and Biology*, pp. 521–548, 2008.

[35] I. Standard, «61000-4-27: testing and measurement techniques-unbalance», *Induction Motormunity Test*, 2000.

[36] M. Terorde, M. Jordan, H. Wattar, J. Lemke, J. Koch, and D. Schulz, «Implementation of multiplex-switches for power feeder balancing in aircraft», in *Power Engineering Conference (UPEC), 2013 48th International Universities'*, IEEE, 2013, pp. 1–6.

[37] Y.-J. Wang, «Analysis of effects of three-phase voltage unbalance on induction motors with emphasis on the angle of the complex voltage unbalance factor», *Energy conversion, IEEE transactions on*, vol. 16, no. 3, pp. 270–275, 2001.

[38] W. L. Winston, M. Venkataramanan, and J. B. Goldberg, *Introduction to mathematical programming*. Thomson/Brooks/Cole, 2003, vol. 1.

[39] R. Woll, «Effect of unbalanced voltage on the operation of polyphase induction motors», *Industry Applications, IEEE Transactions on*, no. 1, pp. 38–42, 1975.

[40] L. A. Wolsey, *Integer programming*. Wiley New York, 1998, vol. 42.

Appendix A

Power

A.1 Instantaneous Power

The instantaneous power is defined as [27]:

$$p = vi \tag{A.1}$$

In an AC circuit, voltage and current vary in time. The most common case is to consider sinusoidal waves for these values. The expressions for these variables are then given by:

$$v = V_m \cos(\omega t + \theta_v) \tag{A.2}$$
$$i = I_m \cos(\omega t + \theta_i) \tag{A.3}$$

and consequently, the power is:

$$p = V_m I_m \cos(\omega t + \theta_v) \cos(\omega t + \theta_i) \tag{A.4}$$

After some mathematical manipulation using trigonometric identities, the following expression can be obtained [27]:

$$p = \frac{V_m I_m}{2} \cos(\theta_v - \theta_i) + \frac{V_m I_m}{2} \cos(\theta_v - \theta_i) \cos(2\omega t) - \frac{V_m I_m}{2} \sin(\theta_v - \theta_i) \sin(2\omega t) \tag{A.5}$$

Equation (A.5) has three terms that can be rewritten as [27]:

$$p = P + P\cos(2\omega t) - Q\sin(2\omega t) \tag{A.6}$$

where

$$P = \frac{V_m I_m}{2} \cos(\theta_v - \theta_i) \tag{A.7}$$
$$Q = \frac{V_m I_m}{2} \sin(\theta_v - \theta_i) \tag{A.8}$$

P is called *real power*, because it describes the power in a circuit that is transformed from electric energy to nonelectric energy. Q is defined as the reactive power.

Note that P and Q carry the same dimension. Nevertheless, to distinguish between them, we use *watt* (W) for real power and *volt-amp reactive* (VAR) for reactive power.

A.2 POWER FACTOR

The cosine of $\theta_v - \theta_i$ plays an important role in power calculations and is called *power factor*:

$$pf = \cos(\theta_v - \theta_i) \tag{A.9}$$

The value of pf does not tell us the value of $(\theta_v - \theta_i)$, since $\cos(\theta_v - \theta_i) = \cos(\theta_i - \theta_v)$. To completely describe this angle, it is common to give the additional information of *lagging* or *leading*. The first one implies that current lags voltage (inductive load) and the second implies that current leads voltage (capacitive load).

A.3 COMPLEX POWER

Complex Power is the complex sum of real power and reactive power:

$$S = P + jQ \tag{A.10}$$

Although it has the same dimensions, we use *volt-amps* (VA) for complex power to distinguish from real and reactive power.

We can think of P, Q and $|S|$ as sides of a right triangle.

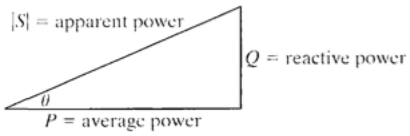


Figure A.1: Complex Power, Real Power and Reactive Power (taken from [27]).

The angle θ can be obtained by:

$$\tan\theta = \frac{Q}{P} = \frac{(V_m I_m/2)\sin(\theta_v - \theta_i)}{(V_m I_m/2)\cos(\theta_v - \theta_i)} = \tan(\theta_v - \theta_i) \tag{A.11}$$

therefore, $\theta = \theta_v - \theta_i$.

The magnitude of complex power:

$$|S| = \sqrt{P^2 + Q^2} \tag{A.12}$$

is defined as the apparent power. The apparent power is an important measure, since it determines the volt-amp capacity necessary to supply the real power.

APPENDIX B

Phasor Notation

A phasor is a complex number based on Euler's identity:

$$\exp^{\pm j\theta} = \cos\theta \pm j\sin\theta, \tag{B.1}$$

and it carries the amplitude and phase angle information of a sinusoidal function [27].

With this identity, cosine and sine functions can be written as

$$\cos\theta = \mathrm{Re}\{e^{\pm j\theta}\} \tag{B.2}$$

$$\sin\theta = \mathrm{Im}\{e^{\pm j\theta}\}, \tag{B.3}$$

where Re and Im refer to the real and imaginary parts of the complex number, respectively.

With this notation, the sinusoidal voltage

$$v = V_m \cos(\omega t + \phi), \tag{B.4}$$

can be expressed as

$$v = V_m \mathrm{Re}\{e^{(\omega t+\phi)}\} \tag{B.5}$$

$$= \mathrm{Re}\{V_m e^{\phi} e^{\omega t}\}. \tag{B.6}$$

In the previous equation, the quantity $V_m e^{\phi}$ is defined as phasor notation. It effectively carries only the amplitude and phase angle of the sinusoidal function. This notation is efficient to deal with situations where the frequency ω is the same for all considered sinusoids. This is commonly the case for alternate current analysis.

An abbreviation is also used:

$$1\angle\phi := 1e^{\phi}. \tag{B.7}$$

This is the angle notation and it is the one used in this thesis.